AFRL-RI-RS-TR-2008-84
Final Technical Report
March 2008

REAL WORLD COGNITIVE MULTI-TASKING AND PROBLEM SOLVING: A LARGE SCALE COGNITIVE ARCHITECTURE SIMULATION THROUGH HIGH PERFORMANCE COMPUTING – PROJECT CASIE

Dartmouth College

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

NOTICE AND SIGNATURE PAGE

AFRL-RI-RS-TR-2008-84 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/s/	/s/
WILLIAM E. MCKEEVER Work Unit Manager	JAMES A. COLLINS, Deputy Chief Advanced Computing Division Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
MAR 2008	Final	Sep 05 – Sep 07

4. TITLE AND SUBTITLE

REAL WORLD COGNITIVE MULTI-TASKING AND PROBLEM SOLVING: A LARGE SCALE COGNITIVE ARCHITECTURE SIMULATION THROUGH HIGH PERFORMANCE COMPUTING - PROJECT CASIE

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA8750-05-2-0284

5c. PROGRAM ELEMENT NUMBER
62702F

6. AUTHOR(S)

Eugene Santos, Jr. and Kiley McEvoy - Dartmouth College
Nael Abu-Ghazaleh and Vinay Kolar , Mark Zhang and Zhen Guo - SUNY Binghamton

5d. PROJECT NUMBER
558B

5e. TASK NUMBER
CA

5f. WORK UNIT NUMBER
S2

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Dartmouth College
8000 Cummings Hall
Hanover NH 03755

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFRL/RITB
525 Brooks Rd
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-RI-RS-TR-2008-84

12. DISTRIBUTION AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED. WPAFB #08-0882

13. SUPPLEMENTARY NOTES

14. ABSTRACT
In its grandest sense, Project CASIE explored the development of a computational system capable of high level perception and problem solving that reflects the cognitive processes of the human brain. Most specifically, it concentrated on better understanding and modeling intuition and insight in a computational fashion. The goal was to address the fundamental problem of modeling and solving "communities" of tasks from a cognitive point of view through multiple problem solving agents working cooperatively or competitively on different subtasks at multiple levels of granularity.

15. SUBJECT TERMS
Cognitive Architecture, Communities of Interest, Problem Solving, Insight

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			William E. McKeever
U	U	U	UL	41	19b. TELEPHONE NUMBER (*Include area code*) 315-330-2897

Table of Contents

List of Figures

List of Tables

Project Goal

Address the fundamental problem of modeling and solving "communities" of tasks from a *cognitive point of view* through multiple problem solving agents working cooperatively or competitively on different subtasks at multiple levels of granularity.

- Agents are naturally grouped into hierarchies or communities, and such groupings may occur dynamically
- Cognition model is based on a strong global coordination mechanism that relies on "focus" in order to elevate a low-power agent into a full-scale "thinking" agent
    - Dynamic redistribution of "brain thinking power"

Project Members

- *Dr. Eugene Santos Jr. & Kiley McEvoy* – Dartmouth College
    - Contributions
        - Model architecture definition
        - Conduction of small scale implementation and testing
- *Dr. Nael Abu-Ghazaleh & Vinay Kolar* - SUNY Binghamton
    - Contributions
        - Multi-agent development
        - Component communication development
        - Large scale deployment
- *Dr. Mark Zhang & Zhen Guo* - SUNY Binghamton
    - Contributions
        - Community Generation Theory development
        - Task relationship identification

Project Summary

In its grandest sense, Project CASIE explored the development of a computational system capable of high level perception and problem solving that reflects the cognitive processes of the human brain. Most specifically, we concentrated on better understanding and modeling intuition and insight in a computational fashion.

The human brain utilizes a wide variety of methods in order to comprehend and solve the various problems faced on a regular basis. Much research has investigated the use of individual mechanisms in single-domain puzzle-type problems, but relatively little work has explored the dynamic use of multiple methods that is required in most real world applications. Advanced abilities such as insight and creativity are inherently used to solve multi-domain problems. Despite the ubiquity of these activities, their inherent mystique and spontaneity render their characterization difficult through conventional methods. This work serves to explore various levels of problem solving as a result of the dynamic utilization of a coordinated set of specialized mechanisms. It is hypothesized that the ability of the mind to dynamically handle complex problems is dependent on the

elegant structure of memory, an overseeing control, and ubiquitous events such as mind-wandering that occur during thought. To demonstrate these theories, a cognitive architecture has been designed through the conflation and further development of current problem solving theories from various research communities. The developed cognitive architecture has been implemented in a computational environment for testing using the real world application of medical diagnosis. Experimental results demonstrate how the coordination of various types of thought, including mind-wandering, can contribute to higher-level problem solving events such as *insight*. It was the ultimate goal of this work to provide a strong foundation for future research in holistic cognitive architectures and high-level problem solving.

While models are successful, they fail to reflect some of the inherent mechanisms of the brain that may be essential to real world problem solving. Many existing models view solving as a goal driven, top down process that is able to work through problems with efficiency and accuracy. They do not include ubiquitous events, such as mind wandering and attention to external stimuli that occur during problem solving. This is due to the notion that such disturbances contain task-unrelated thought [1]. While this may be true, various accounts of insight have shown that complex ideas and solutions can result from mind wandering or in response to external stimuli [2]. In order for this to occur, the mind must be able to dynamically divert attention to thoughts that may be relevant to either active or dormant problems. This ability would require several specialized processes operating simultaneously. It is our hypothesis that productive human cognition is the result of the cooperation between multiple parallel functions, governed by a global coordination mechanism. This hypothesis will be developed through the explanation of theories for the coordinated use of various types of thought as well as their proposed involvement in higher-level forms of problem solving.

The term task-unrelated thought is generally used to describe brain activity not associated with the current goal; for example a day-dream [3]. However, various studies have demonstrated the functional similarities between task-related and task-unrelated thought [4]. Neuroimaging findings show that the patterns of neuron activation during wandering thought strongly overlap those observed during active problem solving. Findings also demonstrate that these two types of thought are proportional to one another. As task demand increases, the evidence of spontaneous unrelated thought decreases [5]. These studies suggest that task-related and task-unrelated thought compete for control of common resources in order to perform their function. Our interpretation, however, is the opposite. We believe that the common resources actually make up a mechanism able to coordinate both types of thought. Furthermore, we feel that the two types of thought are not independent and are both utilized during problem solving. Thus, we reject the notions of related and unrelated and refer to them as *rational* and *intuitive* thought.

The ability to direct one's thought in a goal oriented manner is what allows us to productively interact with our surroundings. Using logic and reason, one is able to make decisions, infer relationships, and manipulate thought. Naturally, these abilities play a large role in problem solving. Upon encountering a problem, one must develop an understanding of the situation and properly select and execute an appropriate strategy. In

common language, the term rational is associated with one's behavior rather than the underlying thought. For this project, it will be used to describe a deliberate thought or action that is "consistent with or based on reason" [6].

Intuition is defined as "the act or faculty of knowing or sensing without the use of rational processes" [6]. Though the term is commonly associated with the spontaneous appearance of thoughts relative to an active problem, we will include the recollection of seemingly irrelevant or irrational thought. When not involved with a computationally intensive task, one may find themselves humming a random song or suddenly recalling a childhood memory. If asked, the source or reason for such thoughts cannot be explained. In some instances, the unrequested thoughts consume all of one's consciousness and can be focused on. Other times, they seem to occur in the background of one's mind, barely perceivable. This feeling is often experienced immediately prior to recalling a necessary bit of information, such as a word to describe a situation. One may have a strong feeling of awareness for a word matching the scenario, but can not immediately verbalize it.

It is within our hypothesis that both problem-relevant and problem-irrelevant intuitions occur through the same mechanism. We suggest that intuitive thought occurs due to associations between concepts on a neurological level. One can agree that a particular stimulus such as the smell of the ocean is capable of eliciting memories of past experiences involving a beach. We will subscribe to the well-supported belief that this is due to memories existing as overlapping neural networks within the brain that host our experiences within their connected structure [7]. Based on the principle of synchronous convergence, networks that are active simultaneously will form a connection and later will be capable of activating one another. In other words, concepts existing in consciousness together will be encoded into memory with an association. Thus, one is prone to develop an association with the smell of the ocean and the visual representation of the beach.

As rational and intuitive thought differ greatly in behavior, the mind would be very limited if only one existed. The abilities of both mechanisms must be coordinated in order for the brain to productively interact with a dynamic environment. We pose that this coordination is managed by an overseeing cognitive mechanism, which will be referred to as meta-cognition. The term has been used by many as a buzzword in experimental education and psychology to describe the ability to stay on task. For the purposes of this project, a more specific definition for meta-cognition will be used, identifying its abilities as "control of learning, planning and selecting strategies, monitoring the progress of learning, correcting errors, analyzing the effectiveness of learning strategies, and changing learning behaviors and strategies when necessary." [8] These abilities will be extended to include control of multiple tasks and monitoring semi-conscious thought. To describe the interaction of meta-cognition, we pose a spectrum of coordination spanning proportional levels of rational and intuitive thought.

A state consisting primarily of rational thought is usually entered following the discovery of a relevant procedure that can now be applied. For example, in working on a long division problem, the method is known and solving the problem is simply a matter of

computation. At this state, meta-cognition has allocated the majority of its attention away from intuitive thought. Studies have shown that during intense task related thought, there is an absence of activity in brain regions associated with monitoring of sensory information [9]. Thus, the brain is less subject to distraction from external stimuli.

When engaged in primarily intuitive thought, meta-cognition allows thought to flow freely in response to concept activations from both external and internal sources. This end of the spectrum is representative of "day-dreaming" or "mind-wandering". For example, one might be reading about insects and begin to think about beetles, followed by a daydream of playing on stage with John Lennon. Though this type of thought may not be working on a particular task, brain regions typically associated with problem solving are occasionally recruited during intuitive thought [4]. It is believed this is to evaluate and retrieve factual information as needed in day-dreaming. This type of free flowing thought often leads into task oriented thought, particularly when the mind encounters a subject of interest.

Collaborative use of rational and intuitive thought occurs when the mind is working towards a complex goal, being one that requires the solving of more than one sub-goal. When a problem is encountered, rational thought is used in attempt to expand and build the problem through logic and reasoning. As rational thought traverses memory, networks will be activated, based on the simultaneous convergence principle. These activations will potentially trigger intuitive thoughts. The extent of intuition is correlated with the activity of rational thought. If rational thought moves quickly through the mind, such as in solving a familiar task, there is less chance for distantly connected networks to become activated, limiting abstractly related solutions. Conversely, if rational thought is working slowly, such as when one is unsure how to solve a problem, distantly related networks have a greater chance of being activated through intuition. As intuitive thought occurs, it is observed by meta-cognition. If the activated concept is thought to be of interest, meta-cognition will redirect the global focus to the newly activated idea.

There are several conflicting views as to what types of problem solving can be classified as insight. Some feel that insight includes suddenly solving a puzzle-type problem while actively attempting it [10]. Within our hypothesis, this is merely a moment of complex intuition preceded by a restructuring of the problem. In our opinion, the fascination with insight is in the ability to unintentionally realize the relation of a current situation to an inactive problem residing in a nearly infinite memory. Thus, we will define insight as the inadvertent realization of the applicability of an idea or situation to a previously unrelated problem that results in a novel and productive integration of the two.

We believe that insight is heavily dependent on mind wandering and the global awareness of one's meta-cognition. While working on a problem, one develops associations and factual links between concepts in memory, allowing the problem to be recalled later in the same manner. Meta-cognition becomes aware of these problems knowing they are of global interest. Thus, if any relation becomes active through intuitive thought, meta-cognition immediately diverts attention to mapping the activation to the problem. Sometimes this recollection might occur following strong activation of a

network directly related to the problem, such as suggested in the Opportunistic Assimilation hypothesis. However, we suggest that activations can occur based on more abstract relations, particularly due to the overlapping of neural networks. For example, for Archimedes, the conceptualization of his body causing the water to overflow may have partially overlapped the existing network containing his problem. Through this activation, intuitive thought could build a perceivable portion of the network, allowing meta-cognition to initiate mapping.

In summary, our hypothesis states that all levels of problem solving occur through the dynamic use of a set of mechanisms whose functions are coordinated by a meta-cognitive component. Rational thought serves to perform cognitive tasks utilizing factual information stored in memory. Traversal of memory networks during such tasks activates related networks causing intuitive thought. These autonomic activations may or may not be perceived depending on the current focus of meta-cognition. When working on a computational intensive task, meta-cognition will focus on management of rational thought and suppression of disruptive thought. In periods of rest, intuitive thought is unrestricted and "mind-wandering" may occur. In solving novel problems, both types of thought are used to develop the problem and discover relevant concepts. Occasionally, a unique traversal path through memory may simultaneously activate two or more previously unrelated networks. Insight is considered to occur if such activation results in a beneficial integration of the networks.

We will now describe our implementation and testing of the CASIE Cognitive Architecture.

CASIE Cognitive Architecture

In order to demonstrate and test the discussed hypothesis for coordinated function, our theoretical mechanisms were integrated into a cognitive architecture. The CASIE architecture is composed of theoretical mechanisms able to process data from a user and cooperatively solve a range of real-world type problems. Medical diagnosis had been chosen as the testing domain due to its wealth of information and manageable data structure. We now explain the logic behind the expression of our theories through a set of architecture components.

Medical diagnosis was selected as a testbed. Selecting a domain in which to test CASIE was inherently difficult. One can design a theoretical architecture capable of handling all types of information understandable by humans. Yet, from a computational standpoint, this completeness would be overly ambitious. Thus, the selected testbed had to be complex enough to be representative of real world problem solving scenarios but also remain adoptable by a computational environment. Four criteria were specified to meet these goals. The main attraction to the use of medical diagnosis was, despite the nearly infinite domain, information used in problem solving could be managed and was readily available.

The CASIE architecture consists of six components operating in parallel to collectively complete tasks through various methods. These components serve to manage the cooperation of uncontrolled and controlled processes involved in problem finding and solving. Attention is dynamically allocated depending on the architecture's state. A visual representation of the architecture can be seen in Figure 1.

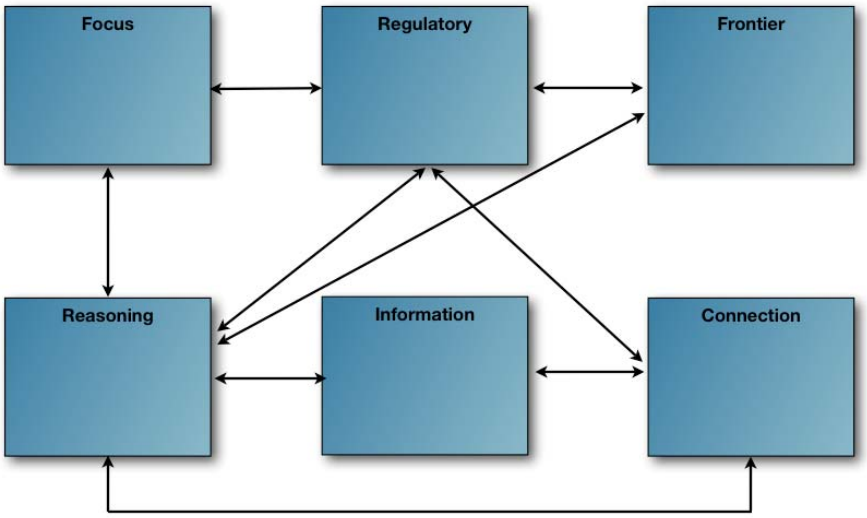

Figure 1: CASIE components and available communication channels

Each component has a specific role in the architecture and its behavior is dependent on the state of the rest of the system. The six components, Information, Reasoning, Connection, Regulatory, Focus, and Frontier, will be briefly explained through descriptions of basic function and detailed interaction examples.

The Information component is representative of one's long term memory. It has been designed to host encountered problems, related information, solution procedures, and methods to validate potential solutions. For the domain of medical diagnosis, these data types have been specified into patients, symptoms, diagnoses, and tests. Patients serve as access points to problems. When a doctor is presented with a diagnosis case, related information is gathered. This information primarily includes the patient's symptoms, which are then used to find potential diagnoses. A doctor may then validate their beliefs or gain new information though the use of medical tests.

The data stored in Information is traversed and utilized based on relationships between its entities. These relationships are classified as either factual or associative based on their method of creation. As implied, factual relationships represent information one feels is definitively true. For ease of implementation as well as the aim to make the architecture expandable to other domains, factual links are represented through four single phrases: "has", "is", "can cause", and "tests for." The use of these phrases is outlined in Table 1.

Table 1: Factual relationship phrases

Phrase	Signifies	Used to related	Example
has	Possession	Patients to Diagnoses Patients to Symptoms	John 'has' Fatigue
is	Hierarchal/ Synonymical	Symptoms to Symptoms Diagnoses to Diagnoses	Lung Cancer 'is' Cancer Tiredness 'is' Fatigue
can cause	Cause & Effect	Diagnoses to Symptoms	Flu 'can cause' Fever
tests for	Solution validation	Tests to Diagnoses	MRI 'tests for' Tumor

The second type of relationship in Information serves to represent links within data created through environment interaction and processing. This type of relation utilizes the word "association" to signify a relationship between entities. Within CASIE, associative relationships are created between simultaneously active entities, based upon the aforementioned principle of synchronous convergence. A diagram of the CASIE memory structure can be seen in Figure 2.

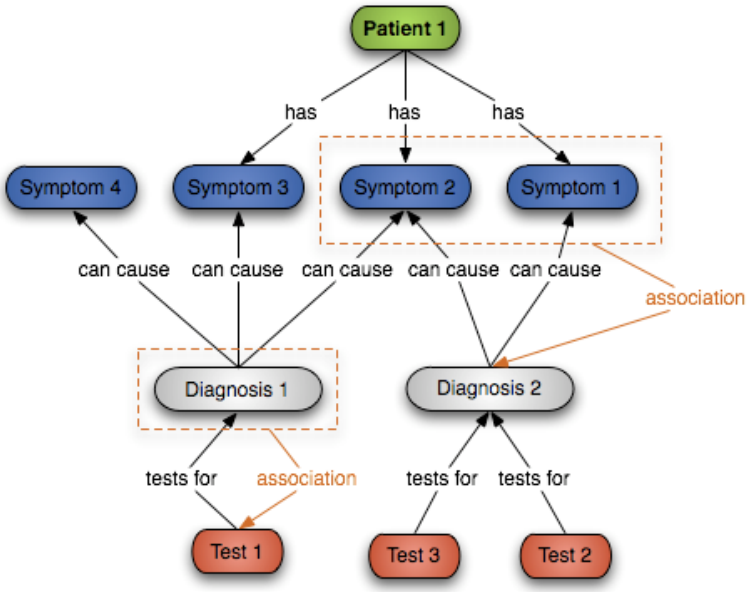

Figure 2: Visual representation of CAISE memory

The Reasoning component is responsible for carrying out tasks associated with rational thought. As a task is worked on, Reasoning temporarily hosts active task knowledge within what will be referred to as the whiteboard. Based on contents of the whiteboard, Reasoning selects appropriate procedures to advance towards a goal. These procedures can include recalling related data from Information, dividing tasks into subtasks, making decisions, and requesting activity from other components. To perform these procedures, Reasoning utilizes factual relationships between knowledge in Information. Reasoning is also responsible for the direct manipulation and addition of such knowledge.

The processes of the Connection component are representative of intuitive thought. Based on the contents of the whiteboard within Reasoning, Connection continuously retrieves associated knowledge from Information. If the system is moving rapidly from task to task, Connection is only capable of finding associations immediately related to the task's domain. However, when a task cannot be solved or one is not active, Connection is able to seek deeper associations.

The Frontier component is responsible for managing CASIE's interaction with the outside world. When new information enters the system, Frontier makes it available to the other components. Similarly, any information that must be expressed internally is presented through Frontier. At this implementation, Frontier has been developed to handle textual information. However, more advanced versions of the component could be representative of a more complete sensory system, including auditory and visual processing as well as mechanical action.

The Focus component serves to maintain a list of active and inactive problems. While working on a task that involves multiple subtasks, Focus hosts the list of jobs that must be done to reach the overall goal. Additionally if tasks are interrupted or CASIE reaches an impasse, tasks are stored in Focus as dormant tasks to be attempted later.

The Regulatory component serves to manage the behavior of CASIE from a global perspective. Its function is analogous to the concept of meta-cognition. When new tasks are encountered, Regulatory determines whether or not the incoming task should interrupt the current activity of the system. When engaged in a task, Regulatory monitors the system activity to ensure that all components are working towards the global goal. Additionally, if any localized activity seems to relate to either the task at hand or a dormant task in Focus, Regulatory will shift attention to investigate the use of that thought. During times of inactivity, Regulatory recalls unsolved tasks from Focus to be re-attempted.

To achieve problem solving ability CASIE's components work cooperatively. The various interactions are outlined in Table 1 and their use is detailed through the examples following.

Table 2: Component Interactions

Components	Interaction
Frontier & Regulatory	As commands and information enter the system, Frontier passes them to Regulatory to handle them. Regulatory also reports system status to Frontier.
Frontier & Reasoning	While learning, information is sent from Frontier to Reasoning for storage. External actions are requested by Reasoning through Frontier.
Regulatory & Reasoning	Regulatory sends commands to Reasoning and observes its activity.
Focus & Reasoning	When tasks are interrupted, or an impasse is reached, Reasoning sends tasks to Focus. Reasoning also passes sub-tasks to Focus.
Focus & Regulatory	Regulatory uses the dormant task list within Focus to determine activity during periods of inactivity.
Reasoning & Information	Reasoning recalls knowledge from Information using factual relationships. While learning, or inference, Reasoning stored knowledge in Information.
Connection & Information	Connection recalls knowledge from Information using association relationships.

In CASIE, insight moments occur when Regulatory realizes the application of current information to the solution of a dormant problem residing in Focus. Upon reaching an impasse, the task as well as all of its failed subtasks are stored in Focus. Additionally associations between the symptom set and the patient are created, based on the principle of synchronous convergence. If during subsequent thought, the association becomes active in Connection, Regulatory interrupts the system and commands Reasoning to attempt to apply the newly learned information. For example, when diagnosing the patient 'Sandy', shown in Figure 3a, an impasse is reached after expanding the two symptoms as much as possible. The case is stored in Focus as an unsolved task and an association is created between 'Skin Symptom + Neurological Symptom' and 'Sandy'. When diagnosing the patient 'Vinny', shown in Figure 3b, Reasoning begins to expand the symptom set. Ordinarily, these expansions would be disregarded and most likely not ever perceived, as Connection discovers an association with the diagnosis 'Neurofibromatosis'. However as 'Mass in Spinal Cord' and 'Tan Skin Patches' expand to 'Neurological Symptom' and 'Skin Symptom' respectively, Connection also discovers the association to the unsolved case of 'Sandy'. Upon this realization, Regulatory instructs Reasoning to attempt to map the analog case to the base. Reasoning determines that Vinny's diagnosis is capable of causing Sandy's symptoms and the diagnosis is validated through a test, shown in Figure 3c.

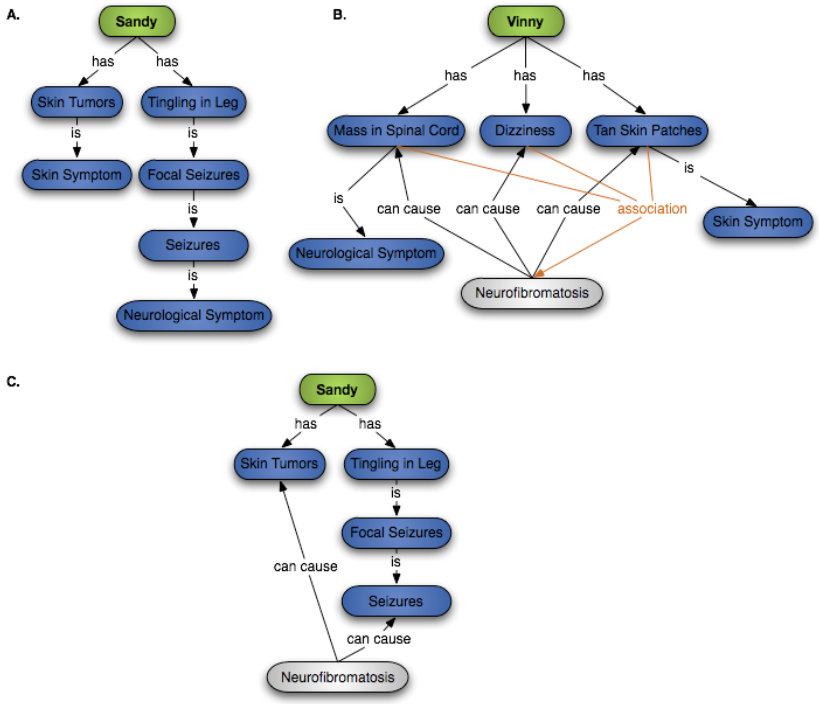

Figure 3: A. First diagnosis leads to impasse, as no associations exist. B. Expansion of another patient's symptoms leads to association with 'Neurofibromatosis'. C. Following insight moment, 'Neurofibromatosis' is validated as a potential diagnosis

Implementation of the CASIE architecture into a computer environment occurred as a collaborative effort between two teams. The majority of the infrastructure and component communication was developed by the team at SUNY Binghamton, while the cognitive algorithms were done by the Dartmouth team. Cougaar, a JAVA software architecture that allows for building distributed agent based applications, was selected as the platform for CASIE's development. Computer implementation allowed for demonstration of the aforementioned theories through experiments using the medical diagnosis testbed.

Synopsis of Experiments

The main theme in our hypothesis is that insight occurs as a result of the monitoring of autonomic intuitive thought through meta-cognition. To demonstrate this theory, several tests were conducted in attempt to trigger an insightful moment within CASIE. Tests consisted of a target problem designed to reach an impasse, and a base problem which could be solved given the contents of CASIE's memory. In each test CASIE was first presented with the target problem. Either immediately following or after intermediate cases, the base problem was presented. Three sets of scenarios were designed to demonstrate the various levels of insight. One set involved a base problem which would directly overlap a portion of the target problem structure. Successful diagnosis of the target problem when solving the base, would demonstrate the spontaneous recollection of

an unsolved problem based on congruent surface attributes. In a second set of scenarios, expansions of the entities from the base problem would coincide with those of the target problem. Successful diagnosis of the target problem in this case, would demonstrate that the inadvertent activation of factually related networks is capable of triggering insight. A third set of tests involved base and target entities with similar but not exact matches, such as 'Lung Complication' and 'Lung Disease'. Successful diagnosis in this case would demonstrate that the partial activation of networks can trigger insight.

The results from our testbed experiments were congruent with our expectations. Successful diagnosis of first type of scenarios demonstrated that through direct activation of entities from an unsolved problem, the problem could be recalled through meta-cognitive processes and subsequently reattempted using newly learned information. We believe this to be a weak form of insight as it utilizes the channels of intuition and meta-cognition however activation of the exact problem components is required. As these full activations would be conscious, it is predicted that the solver would be capable of explaining the train of thought that led to the moment of realization, which counters our definition of insight. The second set of scenarios demonstrates a process closer to our definition of insight, in which the unsolved problem is recalled through automatic activation of related entities. In these cases, such activations may or may not be conscious depending on the attention of meta-consciousness. The scenarios from this experiment represent cases that require a full activation of an entities network. However, the ultimate form of insight has been described as only requiring partial activation of a network for realization to occur. This form was represented in the third set of scenarios. As expected, CASIE was unable to solve these types of cases due to the limitation of textual memory. Demonstration of this type of insight would require true distributed memory.

Conclusions

It is hard to argue that the human brain is not an advanced organ of extensive capabilities. Most are fascinated that a three-pound mass of organic material is able to compose artistic masterpieces and develop advanced scientific theories. Even the simple task of deciding what to eat for dinner is somewhat intriguing. The brain can deal with a wide range of tasks using various methods. Much effort has gone into determining how the brain is able to solve problems at particular levels, but few have ventured to explain all levels of problem solving through the use of common resources in the mind. This work has attempted this task through the presentation of a theoretical cognitive architecture. Our findings demonstrate that all levels of problem solving can be based on various levels of coordination between specialized mechanisms operating in parallel. Rather than a result of search speed, extensive abilities can result from elegance of storage, automatic activation of concepts, and global management.

The current computational version of the CASIE architecture serves to demonstrate the functionality of our primary theories. However, implementation of several other functions is required to fully exploit the power of the architecture. Future efforts could include the addition of learning capability through both inference and experience.

Following this addition, CASIE will be able to internally manipulate the data stored in Information. Following a complex diagnosis, associations would be created between elements of the problem structure. Such associations would aid in the future diagnosis of related problems. Other additions include decision making capability. This would allow CASIE to handle more realistic scenarios in which symptoms associate with multiple diagnoses. Determining which to investigate first would depend on congruence with the symptom set and diagnosis severity.

Following the addition of learning and decision making, the CASIE architecture would be suitable for the addition of advanced abilities. As realized throughout development, higher-level forms of problem solving and creativity are highly dependent on a wealth of interrelated information across many domains. It is hypothesized that by exposing CASIE to a large source of searchable information, the creative ability and occurrence of insight would be significantly increased. Unfortunately the task of manually developing a bounded knowledge base is not only a laborious task; it also defeats the purpose of developing a system able to apply its perceptions to stored problems. Thus, CASIE would require a module allowing it to acquire information as easily as humans. As some readers may have noticed, the structure within the Information component strongly resembles proposed structures of the semantic web, which is foreseen as the next implementation of the internet. The semantic, or machine searchable web, would allow CASIE to gain factual information from a nearly infinite textual source. Currently CASIE is limited to gaining information through a human user. If an impasse is reached, the system cannot seek additional information as a real person is able to do. Using a semantic web interface, CASIE would be capable of learning new symptoms, diagnoses and tests as well as their relationships.

Selected Publications

Kiley McEvoy, "Project CASIE: Cognitive Architecture Studies, Implementations, and Experiments," MS Thesis, Thayer School of Engineering, Dartmouth College, 2007.

Bo Long, Zhongfei (Mark) Zhang, and Philip S. Yu, "Relational Clustering by Symmetric Convex Coding," *Proc. the 24th Annual International Conference on Machine Learning*, Oregon State University, OR, USA, June, 2007.

Bo Long, Zhongfei (Mark) Zhang, and Philip S. Yu, "Graph Partitioning Based on Link Distribution," *Proc. the 22nd Annual Conference on Artificial Intelligence (AAAI-07),* Vancouver, British Columbia, Canada, July, 2007.

Bo Long, Zhongfei (Mark) Zhang, and Philip S. Yu, "A Probabilistic Framework for Relational Clustering," *Proc. the 13th ACM International Conference on Knowledge Discovery and Data Mining*, San Jose, CA, USA, August, 2007.

References

1. Giambra, L.M., A Laboratory Method for Investigating Influences on Switching Attention to Task-Unrelated Imagery and Thought. *Consciousness and Cognition*, 1995. 4.
2. Andreasen, N., *The Creating Brain*. 2005, New York, NY: The Dana Foundation.
3. Smallwood, J., M. Obonsawin, and D. Heim, Task unrelated thought: the role of distributed processing. *Consciousness and Cognition*, 2003. 12(2).
4. Christoff, K., J.M. Ream, and J.D.E. Gabrieli, Neural Basis of Spontaneous Thought Processes. *Cortex*, 2004. 40.
5. Teasdale, J.D., et al., Working memory and stimulus-independent thought: Effects of memory load and presentation rate. *European Journal of Cognitive Psychology*, 1993. 5: p. 417-433.
6. *The American Heritage® Dictionary of the English Language*, Fourth Edition. 2004, Houghton Mifflin Company.
7. Fuster, J.n.M., Network Memory. *Trends in Neuroscience*, 1997. 20(10).
8. Ridley, S., et al., Self-regulated learning: the interactive influence of metacognitive awareness and goal-setting. *Journal of Experimental Education*, 1992. 60(4).
9. Raichle, M.E., The neural correlates of consciousness: an analysis of cognitive skill learning. *Philosophical Transactions of the Royal Society of London*, 1998. 353: p. 1889-1901.
10. Bowden, E.M., et al., New approaches to demystifying insight. *Trends in Cognitive Sciences*, 2005. 9(7).

APPENDIX A: Graph Partitioning Based on Link Distributions

Bo Long and **Mark (Zhongfei) Zhang**
Computer Science Dept., SUNY Binghamton
Binghamton, NY 13902
{blong1, zzhang}@binghamton.edu

Philip S. Yu
IBM Watson Research Center
19 skyline Drive, Hawthorne, NY 10532
psyu@us.ibm.com

Abstract

Existing graph partitioning approaches are mainly based on optimizing edge cuts and do not take the distribution of edge weights (link distribution) into consideration. In this paper, we propose a general model to partition graphs based on link distributions. This model formulates graph partitioning under a certain distribution assumption as approximating the graph affinity matrix under the corresponding distortion measure. Under this model, we derive a novel graph partitioning algorithm to approximate a graph affinity matrix under various Bregman divergences, which correspond to a large exponential family of distributions. We also establish the connections between edge cut objectives and the proposed model to provide a unified view to graph partitioning.

Introduction

Graph partitioning is an important problem in many machine learning applications, such as circuit partitioning, VLSI design, task scheduling, bioinformatics, and social network analysis. Existing graph partitioning approaches are mainly based on edge cut objectives, such as Kernighan-Lin objective (Kernighan & Lin 1970), normalized cut (Shi & Malik 2000), ratio cut (Chan, Schlag, & Zien 1993), ratio association(Shi & Malik 2000), and min-max cut (Ding *et al.* 2001).

The main motivation of this study comes from the fact that graphs from different applications may have very different statistical characteristics for their edge weights. Specifically, the graphs may have very different link distributions, where the link distribution refers to the *distribution of edge weights* in a graph. For example, in a graph with binary weight edges, the link distribution can be modeled as a Bernoulli distribution; in a graph with edges of real value weights, the link distribution may be modeled as an exponential distribution or a normal distribution. This fact naturally raises the following questions: is it appropriate to use edge cut objectives for all kinds of graphs with different link distributions? If not, what kinds of graphs the edge cut objectives work well for? How to make use of link distributions to partition different types of graphs? This paper attempts to answer these questions.

Another motivation of this study is to derive an effective algorithm to improve the existing graph partitioning algorithms on some aspects. For example, the popular spectral approaches involve expensive eigenvector computation and extra post-processing on eigenvectors to obtain the partitioning; the multi-level approaches such as METIS (Karypis & Kumar 1998) restrict partitions to have an equal size.

In this paper, we propose a general model to partition graphs based on link distributions. The key idea is that by viewing the link distribution of a graph as a mixture of link distributions within and between different partitions, we can learn the mixture components to find the partitioning of the graph. The model formulates partitioning a graph under a certain distribution assumption as approximating the graph affinity matrix under the corresponding distortion measure. Second, under this model, we derive a novel graph partitioning algorithm to approximate a graph affinity matrix under various Bregman divergences, which correspond to a large exponential family distributions. Our theoretic analysis and experiments demonstrate the the potential and effectiveness of the proposed model and algorithm. Third, we also establish the connections between the proposed model and the edge cut objectives to provide a unified view to graph partitioning.

We use the following notations in this paper. Capital letters such as A, B and C denote matrices; A_{ij} or $[A]_{ij}$ denote the (i,j)th element in A; small boldface letters such as $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{c}$ denote column vectors. A graph is denoted by $G = (\mathcal{V}, \mathcal{E}, A)$, which is made up of a set of vertices $\mathcal{V}$ and a set of edges $\mathcal{E}$, and the affinity matrix A of dimension $|\mathcal{V}| \times |\mathcal{V}|$, whose entries represent the weights of the edges.

Related Work

Graph partitioning divides a graph into subgraphs by finding the best edge cuts of the graph. Several edge cut objectives, such as the average cut (Chan, Schlag, & Zien 1993), average association (Shi & Malik 2000), normalized cut (Shi & Malik 2000), and min-max cut (Ding *et al.* 2001), have been proposed. Various spectral algorithms have been developed for these objective functions (Chan, Schlag, & Zien 1993; Shi & Malik 2000; Ding *et al.* 2001). These algorithms use the eigenvectors of a graph affinity matrix, or a matrix derived from the affinity matrix, to partition the graph.

Multilevel methods have been used extensively for graph

partitioning with the Kernighan-Lin objective, which attempts to minimize the cut in the graph while maintaining equal-sized clusters (Bui & Jones 1993; Hendrickson & Leland ; Karypis & Kumar 1998). In multilevel algorithms, the graph is repeatedly coarsened level by level until only a small number of nodes are left. Then, an initial partitioning on this small graph is performed. Finally, the graph is uncoarsened level by level, and at each level, the partitioning from the previous level is refined using a refinement algorithm.

Recently, graph partitioning with an edge cut objective has been shown to be mathematically equivalent to an appropriately weighted kernel k-means objective function (Dhillon, Guan, & Kulis 2004; 2005). Based on this equivalence, the weighted kernel k-means algorithm has been proposed for graph partitioning (Dhillon, Guan, & Kulis 2004; 2005). Yu, Yu, & Tresp (2005) propose graph-factorization clustering for the graph partitioning, which seeks to construct a bipartite graph to approximate a given graph. Long *et al.* (2006) propose a framework of relation summary network to cluster K-partite graphs.

Another related field is unsupervised learning with Bregman divergences (S.D.Pietra 2001; Wang & Schuurmans 2003). Banerjee *et al.* (2004b) generalizes the classic k-means to Bregman divergences. A generalized co-clustering framework is presented by Banerjee *et al.* (2004a) wherein any Bregman divergence can be used in the objective function.

Model Formulation

We first define the link distribution as the follows.

Definition 1. *Given a graph $G = (\mathcal{V}, \mathcal{E}, A)$, the link distribution $f_{\mathcal{V}_1\mathcal{V}_2}$ is the probability density of edge weights between nodes in $\mathcal{V}_1$ and $\mathcal{V}_2$, where $\mathcal{V}_1, \mathcal{V}_2 \subseteq \mathcal{V}$.*

Based on Definition 1, the link distribution for the whole graph G is $f_{\mathcal{V}\mathcal{V}}$. The model assumption is that if G has k disjoint partitions $\mathcal{V}_1, \ldots, \mathcal{V}_k$, then $f_{\mathcal{V}\mathcal{V}} = \sum_{1\le i\le j\le k} \pi_{ij} f_{\mathcal{V}_i\mathcal{V}_j}$, where π_{ij} is the mixing probability such that $\sum_{1\le i\le j\le k} \pi_{ij} = 1$. Basically, the assumption states that the link distribution of a graph is a mixture of the link distributions within and between partitions. The intuition behind the assumption is that the vertices within the same partition are related in a (statistically) similar way to each other and the vertices from different partitions are related in different ways to each other from those within the same partition. In Section 5, we show that the traditional edge cut objectives also implicitly make this assumption under a normal distribution with extra constraints.

Let us have an illustrative example. Figure 1(a) shows a graph of six vertices and seven unit weight edges. It is natural to partition the graph into two components, $\mathcal{V}_1 = \{v_1, v_2, v_3\}$ and $\mathcal{V}_2 = \{v_4, v_5, v_6\}$. The link distribution of the whole graph can be modeled as a Bernoulli distribution $f_{\mathcal{V}\mathcal{V}}(x; \theta_{\mathcal{V}\mathcal{V}})$ with the parameter $\theta_{\mathcal{V}\mathcal{V}} = \frac{7}{15}$ (the number of edges in the graph is 7 and the number of possible edges is 15). Similarly, the link distributions for edges within and between $\mathcal{V}_1$ and $\mathcal{V}_2$ are Bernoulli distributions, $f_{\mathcal{V}_1\mathcal{V}_1}(x; \theta_{\mathcal{V}_1\mathcal{V}_1})$ with $\theta_{\mathcal{V}_1\mathcal{V}_1} = 1$, $f_{\mathcal{V}_2\mathcal{V}_2}(x; \theta_{\mathcal{V}_2\mathcal{V}_2})$ with

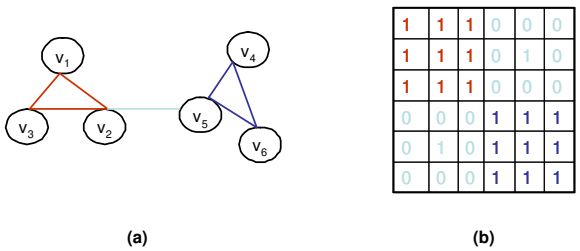

Figure 1: A graph with two partitions (a) and its graph affinity matrix (b).

$\theta_{\mathcal{V}_2\mathcal{V}_2} = 1$, and $f_{\mathcal{V}_1\mathcal{V}_2}(x; \theta_{\mathcal{V}_1\mathcal{V}_2})$ with $\theta_{\mathcal{V}_1\mathcal{V}_2} = \frac{1}{9}$. Note that $f_{\mathcal{V}\mathcal{V}}$ is a mixture of $f_{\mathcal{V}_1\mathcal{V}_1}$, $f_{\mathcal{V}_2\mathcal{V}_2}$ and $f_{\mathcal{V}_1\mathcal{V}_2}$, which can be verified by $\theta_{\mathcal{V}\mathcal{V}} = \frac{3}{15}\theta_{\mathcal{V}_1\mathcal{V}_1} + \frac{9}{15}\theta_{\mathcal{V}_1\mathcal{V}_2} + \frac{3}{15}\theta_{\mathcal{V}_2\mathcal{V}_2}$ (the mixing probability for $f_{\mathcal{V}_1\mathcal{V}_2}$, $\frac{9}{15}$, follows the fact that the number of possible edges between $\mathcal{V}_1$ and $\mathcal{V}_2$ is 9; similarly for other proportion probabilities).

Learning mixture components of the link distribution of a graph is much more difficult than learning a traditional mixture model, since the graph structure needs to be considered, i.e., our goal is to find the mixture components associated with subgraphs and not just to simply draw the similar edges from anywhere in the graph to form a component. For example, in Figure 1(a), without considering the graph structure, the edge weights from two partitions $\mathcal{V}_1$ and $\mathcal{V}_2$ cannot be separated. To tackle this difficulty, we model the problem based on the graph affinity matrix, which contains all the information for a graph.

Figure 1(b) shows the graph affinity matrix for the graph in Figure 1(a). We observe that if the vertices within the same partition are arranged together, the edge weights within and between partitions form the diagonal blocks and off-diagonal blocks, respectively. Hence, learning the link distribution in a graph is equivalent to learning different distributions for non-overlapping blocks in the graph affinity matrix. To estimate the sufficient statistic for each block, we need to solve the problem of likelihood maximization. It is shown that maximizing likelihood under a certain distribution corresponds to minimizing distance under the corresponding distortion measure (Collins, Dasgupta, & Reina 2001). For example, the normal distribution, Bernoulli distribution, multinomial distribution and exponential distribution correspond to Euclidean distance, logistic loss, KL-divergence and Itakura-Satio distance, respectively. Therefore, learning the distributions of the blocks in a graph affinity matrix can be formulated as approximating the affinity matrix under a certain distortion measure. Formally, we define graph partitioning as the following optimization problem of matrix approximation.

Definition 2. *Given a graph $G = (\mathcal{V}, \mathcal{E}, A)$ where $A \in \mathbb{R}^{n\times n}$, a distance function $\mathfrak{D}$, and a positive integer k, the optimized partitioning is given by the minimization,*

$$\min_{C\in\{0,1\}^{n\times k}, B\in\mathbb{R}^{k\times k}} \mathfrak{D}(A, CBC^T), \tag{1}$$

where $C \in \{0,1\}^{n\times k}$ is an indicator matrix such that $\sum_j C_{ij} = 1$, i.e., $C_{ij} = 1$ indicates that the ith vertex belongs to the jth partition, and $\mathfrak{D}$ is a separable distance function such that $\mathfrak{D}(X, Y) = \sum_{i,j} \mathfrak{D}(X_{ij}, Y_{ij})$.

We call the model in Definition 2 as the Graph Partitioning with Link Distribution (GPLD). GPLD provides not only the partitioning of the given graph, which is denoted by the *partition indicator matrix* C, but also the *partition representative matrix* B, which consists of the sufficient statistics for edge weights within and between partitions. For example, $B = \begin{bmatrix} 1 & 1/9 \\ 1/9 & 1 \end{bmatrix}$ for the example in Fig 1(b). B also provides an intuition about the quality of the partitioning, since the larger the difference between the diagonal and the off-diagonal elements, the better the partitions are separated. Note that GPLD does not restrict A to be symmetric or non-negative. Hence, it is possible to apply GPLD to directed graphs or graphs with negative weights, though in this paper our main focus is undirected graphs with non-negative weights.

Algorithm Derivation

First we derive an algorithm for GPLD model based on the most popular distance function, Euclidean distance function. Under Euclidean distance function, our task is

$$\min_{C\in\{0,1\}^{n\times k}, B\in\mathbb{R}^{k\times k}} ||A - CBC^T||^2. \tag{2}$$

We prove the following theorem which is the basis of our algorithm.

Theorem 3. *If $C \in \{0,1\}^{n\times k}$ and $B \in \mathbb{R}_+^{k\times k}$ is the optimal solution to the minimization in* (2)*, then*

$$B = (C^TC)^{-1}C^TAC(C^TC)^{-1}. \tag{3}$$

Proof. The objective function in Definition 2 can be expanded as follows.

$$\begin{aligned} L &= ||A - CBC^T||^2 \\ &= \text{tr}((A - CBC^T)^T(A - CBC^T)) \\ &= \text{tr}(A^TA) - 2\text{tr}(CBC^TA) + \text{tr}(CBC^TCBC^T) \end{aligned}$$

Take the derivative with respect to B, we obtain

$$\frac{\partial L}{\partial B} = -2C^TBC + 2C^TCBC^TC. \tag{4}$$

Solve $\frac{\partial L}{\partial B} = 0$ to obtain

$$B = (C^TC)^{-1}C^TAC(C^TC)^{-1}; \tag{5}$$

This completes the proof of the theorem. □

Based on Theorem 3, we propose an alternative optimization algorithm, which alternatively updates B and C until convergence. We first fix C and update B. Eq (3) in Theorem 3 provides an updating rule for B,

$$B = (C^TC)^{-1}C^TAC(C^TC)^{-1}. \tag{6}$$

This updating rule can be implemented more efficiently than it appears. First, it does not really involve computing inverse matrices, since C^TC is a special diagonal matrix with the size of each cluster on its diagonal such that $[C^TC]_{pp} = |\pi_p|$, where $|\pi_p|$ denotes the size of the pth partitioning; second, the product of C^TAC can be calculated without normal matrix multiplication, since C is an indicator matrix.

Then, we fix B and update C. Since each row of C is an indicator vector with only one element equal to 1, we adopt the re-assignment procedure to update C row by row. To determine which element of the hth row of C is equal to 1, for $p = 1, \ldots, k$, each time we let $C_{hp} = 1$ and compute the objective function $L = ||A - CBC^T||^2$, which is denoted as L_p, then

$$C_{hp^*} = 1 \text{ for } p^* = \arg\min_p L_p \tag{7}$$

Note that when we update the hth row of C, the necessary computation involves only the hth row or column of A and CBC^T.

Therefore, updating rules (6) and (7) provide a new graph partitioning algorithm, GPLD under Euclidean distance.

Presumably for a specific distance function used in Definition 2, we need to derive a specific algorithm. However, a large number of useful distance functions, such as Euclidean distance, generalized I-divergence, and KL divergence, can be generalized as the Bregman divergences (S.D.Pietra 2001; Banerjee *et al.* 2004b), which correspond to a large number of exponential family distributions. Moreover, the nice properties of Bregman divergences make it easy to generalize updating rules (6) and (7) to all Bregman divergences. The definition of a Bregman divergence is given as follows.

Definition 4. *Given a strictly convex function, $\phi : S \mapsto \mathbb{R}$, defined on a convex set $S \subseteq \mathbb{R}^d$ and differentiable on the interior of S, int(S), the Bregman divergence $D_\phi : S \times$ int$(S) \mapsto [0, \infty)$ is defined as*

$$D_\phi(x, y) = \phi(x) - \phi(y) - (x - y)^T\nabla\phi(y), \tag{8}$$

where $\nabla\phi$ is the gradient of ϕ.

Table 1 shows a list of popular Bregman divergences and their corresponding Bregman convex functions. The following Theorem provide an important property of Bregman divergence.

Theorem 5. *Let X be a random variable taking values in $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n \subset S \subseteq \mathbb{R}^d$ following v. Given a Bregman divergence $D_\phi : S \times$ int$(S) \mapsto [0, \infty)$, the problem*

$$\min_{s\in S} E_v[D_\phi(X, s)] \tag{9}$$

has a unique minimizer given by $s^* = E_v[X]$.

The proof of Theorem 5 is omitted (please refer (S.D.Pietra 2001; Banerjee *et al.* 2004b)). Theorem 5 states that the Bregman representative of a random variable is always the expectation of the variable. Hence, when given a sample of a random variable, the optimal estimation of the Bregman representative is always the mean of the sample. Under the GPLD model, B_{pq} is the Bregman representative of each block of an affinity matrix. When C is given, i.e., the membership of each block is known, according to Theorem 5, B_{pq} is obtained as the mean of each block,

$$B_{pq} = \frac{1}{|\pi_p||\pi_q|} \sum_{i\in\pi_p, j\in\pi_q} A_{ij}, \tag{10}$$

Table 1: A list of Bregman divergences and the corresponding convex functions.

Name	$D_\phi(x, y)$	$\phi(x)$	Domain
Euclidean distance	$\|\|\mathbf{x}-\mathbf{y}\|\|^2$	$\|\|\mathbf{x}\|\|^2$	$\mathbb{R}^d$
Generalized I-divergence	$\sum_{i=1}^d x_i \log(\frac{x_i}{y_i}) - \sum_{i=1}^d (x_i - y_i)$	$\sum_{i=1}^d x_i \log(x_i)$	$\mathbb{R}_+^d$
Logistic loss	$x \log(\frac{x}{y}) + (1-x)\log(\frac{1-x}{1-y})$	$x\log(x) + (1-x)\log(1-x)$	$\{0, 1\}$
Itakura-Saito distance	$\frac{x}{y} - \log xy - 1$	$-\log x$	$(0, \infty)$
Hinge loss	$\max\{0, -2\text{sign}(-y)x\}$	$\|x\|$	$\mathbb{R} \setminus \{0\}$
KL-divergence	$\sum_{i=1}^d x_i \log(\frac{x_i}{y_i})$	$\sum_{i=1}^d x_i \log(x_i)$	d-Simplex
Mahalanobis distance	$(\mathbf{x}-\mathbf{y})^T\mathbf{A}(\mathbf{x}-\mathbf{y})$	$\mathbf{x}^T\mathbf{A}\mathbf{x}$	$\mathbb{R}^d$

Algorithm 1 Graph Partitioning with Bregman Divergences

Input: A graph affinity matrix A, a Bregman divergence D_ϕ, and a positive integer k.

Output: A partition indicator matrix C and a partition representative matrix B.

Method:

1: Initialize B.
2: **repeat**
3: **for** $h = 1$ to n **do**
4: $C_{hp^*} = 1$ for $p^* = \arg\min_p L_p$ where L_p denotes $D_\phi(A, CBC^T)$ for $C_{hp} = 1$.
5: **end for**
6: $B = (C^TC)^{-1}C^TAC(C^TC)^{-1}$.
7: **until** convergence

where π_p and π_q denote the pth and the qth cluster, respectively, and $1 \le p \le k, 1 \le q \le k, 1 \le i \le n$ and $1 \le j \le n$. If we write Eq (10) in a matrix form, we obtain Eq. (3), i.e., Theorem 3 is true for all Bregman divergences. Hence, updating rule (6) is applicable to GPLD with any Bregamen divergneces. For updating rule (7), there is only a minor change for a given Bregman divergence, i.e., we calculate the object function L based on this given bregman divergence.

Therefore, we obtain a general graph partitioning algorithm, Graph Partitioning with Bregman Divergences (GPBD), which is summarized in Algorithm 1. Unlike the traditional graph partitioning approaches, this simple algorithm is capable of partitioning graphs under different link distribution assumptions by adopting different Bregman divergences. The computational complexity of GPBD can be shown to be $O(tn^2k)$ for t iterations. For a sparse graph, it is reduced to $O(t|\mathcal{E}|k)$. GPBD is faster than the popular spectral approaches, which involve expensive eigenvector computation (typically $O(n^3)$) and extra post-processing on eigenvectors to obtain the partitioning. Comparing with the multi-level approaches such as METIS (Karypis & Kumar 1998), GPBD does not restrict partitions to have an equal size.

The convergence of Algorithm 1 is guaranteed based on the following facts. First, based on Theorem 3 and Theorem 5, the objective function is non-increasing under updating rule (6); second, by the criteria for reassignment in updating rule (7), it is trivial to show that the objective function is non-increasing under updating rule (7).

A Unified View to Graph Partitioning

In this section, we establish the connections between the GPLD model and the edge cut objectives to provide a unified view for graph partitioning.

In general, the edge cut objectives, such as ratio association (Shi & Malik 2000), ratio cut(Chan, Schlag, & Zien 1993), Kernighan-Lin objective (Kernighan & Lin 1970), and normalized cut (Shi & Malik 2000), can be formulated as the following trace maximization (Zha *et al.* 2002; Dhillon, Guan, & Kulis 2004; 2005),

$$\max \text{tr}(\tilde{C}^T A \tilde{C}). \tag{11}$$

In (11), typically $\tilde{C}$ is a weighted indicator matrix such that

$$\tilde{C}_{ij} = \begin{cases} \frac{1}{|\pi_j|^{\frac{1}{2}}} & \text{if } v_i \in \pi_j \\ 0 & \text{otherwise} \end{cases}$$

where $|\pi_j|$ denotes the number of nodes in the jth partition. In other words, $\tilde{C}$ satisfies the constraints $\tilde{C} \in \mathbb{R}_+^{n\times k}$ and $\tilde{C}^T\tilde{C} = I_k$, where I_k is the $k \times k$ identity matrix.

We propose the following theorem to show that the various edge cut objectives are mathematically equivalent to a special case of the GPLD model. To be consistent with the weighted indicator matrix used in edge cut objects, in the following theorem we modify the constraints on C as $C \in \mathbb{R}_+$ and $C^TC = I_k$ to make C to be a weighted indicator matrix.

Theorem 6. *The GPLD model under Euclidean distance function and $B = rI_k$ for $r > 0$, i.e.,*

$$\min_{\substack{C\in\mathbb{R}_+^{n\times k},\\ C^TC=I_k}} ||A - C(rI_k)C^T||^2 \tag{12}$$

is equivalent to the maximization

$$\max \textit{tr}(C^TAC), \tag{13}$$

where tr denotes the trace of a matrix.

Proof. Let L denote the objective function in Eq. 12.

$$\begin{aligned} L &= ||A - rCC^T||^2 \\ &= \text{tr}((A - rCC^T)^T(A - rCC^T)) \\ &= \text{tr}(A^TA) - 2r\text{tr}(CC^TA) + r^2\text{tr}(CC^TCC^T) \\ &= \text{tr}(A^TA) - 2r\text{tr}(C^TAC) + r^2k \end{aligned}$$

The above deduction uses the property of trace $\text{tr}(XY) = \text{tr}(YX)$. Since $\text{tr}(A^TA)$, r and k are constants, the minimization of L is equivalent to the maximization of $\text{tr}(C^TAC)$. The proof is completed. □

Table 2: Summary of the synthetic graphs

Graph	Parameter	n	k	distribution
syn1	$\begin{bmatrix} 3 & 3 & 2.7 \\ 3 & 2.7 & 2.7 \\ 2.7 & 2.7 & 3 \end{bmatrix}$	300	3	Normal
syn2	$\begin{bmatrix} 6.9 & 7 & 6.3 \\ 7 & 6.3 & 6.3 \\ 6.3 & 6.3 & 7 \end{bmatrix}$	600	3	Poisson
syn3	$\mathbb{R}^{20\times 20}$	20000	20	Normal

Theorem 6 states that with the partition representative matrix B restricted to be of the form rI_k, the GPLD model under Euclidean distance is reduced to the trace maximization in (13). Since various edge cut objectives can be formulated as the trace maximization, Theorem 6 establishes the connection between the GPLD model and the existing edge cut objective functions.

Based on this connection, edge cut objectives make two implicit assumptions for a graph's link distribution. First, Euclidean distance in Theorem 6 implies normal distribution assumption for the edge weights in a graph. Second, since the off-diagonal entries in B represent the mean edge weights between partitions and the diagonal elements of B represent the the mean edge weights within partitions, restricting B to be of the form rI_k for $r > 0$ implies that the edges between partitions are very sparse (close to 0) and the edge weights within partitions have the same positive expectation r. However, these two assumptions are not appropriate for the graphs whose link distributions deviate from normal distribution or dense graphs. Therefore, compared with the edge cut based approaches, the GPBD algorithm is more flexible to deal with graphs with different statistic characteristics.

Experimental Results

Although GPBD actually provides a family of algorithms under various Bregman divergences, due to the space limit, in this paper we present the experimental evaluation of the effectiveness of the GPBD algorithm under two most popular divergences, GPBD under Euclidean Distance (GPBD-ED) corresponding to normal distribution, and GPBD under Generalized I-divergence (GPBD-GI) corresponding to Poisson distribution, in comparison with two representative graph partitioning algorithms, Normalized Cut (NC) (Shi & Malik 2000; Ng, Jordan, & Weiss 2001) and METIS (Karypis & Kumar 1998).

We use synthetic data to simulate graphs whose edge weights are under normal and poisson distributions. The distribution parameters to generate the graphs are listed in the second column of Table 2 as matrices. In a parameter matrix P, P_{ij} denotes the distribution parameter that generates the edge weights between the nodes in the ith partition and the nodes in the jth partition. Graph syn3 has twenty partitions of 20000 nodes and about 10 million edges. Due to the space limit, its distribution parameters are omitted here.

The graphs based on the text data have been widely used to test graph partitioning algorithms (Ding *et al.* 2001; Dhillon 2001; Zha *et al.* 2001). In this study, we construct real graphs based on various data sets from the 20-newsgroups (Lang 1995) data, which contains about $20,000$ articles from the 20 news groups and can be used to generate data sets of different sizes, balances and difficulty levels. We pre-process the data by removing stop words and file headers and selecting the top 2000 words by the mutual information. Each document is represented by a term-frequency vector using TF-IDF weights and the cosine similarity is adopted for the edge weight. Specific details of data sets are listed in Table 3. For example, the third row of Table 3 shows that three data sets NG5-1, NG5-2 and NG5-3 are generated by sampling from five newsgroups with size 900, 1200 and 1450, respectively, and with *balance* 1.5, 2.5, and 4, respectively. Here *balance* denotes the ratio of the largest partition size to the smallest partition size in a graph. Normalized Mutual Information (NMI) (Strehl & Ghosh 2002) is used for performance measure, which is a standard way to measure the cluster quality. The final performance score is the average of twenty runs.

Table 4 shows the NMI scores of the four algorithms. For the synthetic data syn1 and syn3 with normal link distribution, the GPBD-ED algorithm, which assumes normal distribution for the links, provides the best NMI score. Similarly, for data syn2 with poisson link distribution, the GPBD-GI algorithm, which assumes poisson distribution for the links, provides the best performance.

For real graphs, we observe that GPBD-GI provides best NMI scores for all the graphs and preforms significantly better than NC and METIS in most graphs . This implies that link distributions of the graphs are closer to Poisson distribution than normal distribution. How to determine appropriate link distribution assumption for a given graph is beyond the scope of this paper. However, the result shows that the appropriate link distribution assumption (appropriate distance function for GPBD) leads to a significant improvement on the partitioning quality. For example, for the graph NG2-3, even NC totally fails and other algorithms perform poorly, GPBD-IS still provides satisfactory performance. We observe that all the algorithms perform poorly for NG10. One possible reason for this is that in NG10 some partitions are heavily overlapped and very unbalanced. We also observe that the performance of the GPBD with the appropriate distribution is more robust to unbalanced graphs. For example, from NG2-1 to NG2-3, the performance of GPBD-IS decreases much less than those of NC and METIS. One possible reason for METIS's performance deterioration on unbalanced graphs is that it restricts partitions to have equal size.

Conclusion

In this paper, we propose a general model to partition graphs based on link distribution. This model formulates graph partitioning under a certain distribution assumption as approximating the graph affinity matrix under the corresponding distortion measure. Under this model, we derive a novel graph partitioning algorithm to approximate a graph affinity matrix under various Bregman divergences, which correspond to a large exponential family of distributions. Our theoretic analysis and experiments demonstrate the potential

Table 3: Subsets of Newsgroup Data for constructing graphs.

Name	**Newsgroups Included**	**# Documents**	**Balance**
NG2-1/2/3	alt.atheism, comp.graphics	330/525/750	1.2/2.5/4
NG3-1/2/3	comp.graphics, rec.sport.hockey,talk.religion.misc	480/675/900	1.2/2.5/4
NG5-1/2/3	comp.os.ms-windows.misc, comp.windows.x, rec.motorcycles,sci.crypt, sci.space	900/1200/1450	1.5/2.5/4
NG10	comp.graphics, comp.sys.ibm.pc.hardware, rec.autos, rec.sport.baseball,sci.crypt, sci.med,comp.windows.x, soc.religion.christian, talk.politics.mideast,talk.religion.misc	5600	7

Table 4: NMI scores of the five algorithms

Data	NC	METIS	GPBD-ED	GPBD-GI
syn1	0.673 ± 0.081	0.538 ± 0.016	$\mathbf{0.915 \pm 0.017}$	0.893 ± 0.072
syn2	0.648 ± 0.052	0.533 ± 0.018	0.828 ± 0.139	$\mathbf{0.863 \pm 0.111}$
syn3	0.801 ± 0.029	0.799 ± 0.010	$\mathbf{0.933 \pm 0.047}$	0.811 ± 0.055
NG2-1	0.482 ± 0.299	0.759 ± 0.024	0.678 ± 0.155	$\mathbf{0.824 \pm 0.045}$
NG2-2	0.047 ± 0.041	0.400 ± 0.000	0.283 ± 0.029	$\mathbf{0.579 \pm 0.073}$
NG2-3	0.042 ± 0.023	0.278 ± 0.000	0.194 ± 0.008	$\mathbf{0.356 \pm 0.027}$
NG3-1	0.806 ± 0.108	0.810 ± 0.017	0.718 ± 0.128	$\mathbf{0.852 \pm 0.081}$
NG3-2	0.185 ± 0.116	0.501 ± 0.012	0.371 ± 0.131	$\mathbf{0.727 \pm 0.070}$
NG3-3	0.048 ± 0.013	0.546 ± 0.016	0.235 ± 0.091	$\mathbf{0.631 \pm 0.179}$
NG5-1	0.598 ± 0.077	0.616 ± 0.032	0.550 ± 0.043	$\mathbf{0.662 \pm 0.025}$
NG5-2	0.5612 ± 0.030	0.570 ± 0.020	0.546 ± 0.032	$\mathbf{0.670 \pm 0.022}$
NG5-3	0.426 ± 0.060	0.574 ± 0.018	0.515 ± 0.033	$\mathbf{0.668 \pm 0.035}$
NG10	0.281 ± 0.011	0.310 ± 0.017	0.308 ± 0.015	$\mathbf{0.335 \pm 0.009}$

and effectiveness of the proposed model and algorithm. We also show the connections between the traditional edge cut objectives and the proposed model to provide a unified view to graph partitioning.

Acknowledgement

This work is supported in part by NSF (IIS-0535162), AFRL (FA8750-05-2-0284), and AFOSR (FA9550-06-1-0327).

References

Banerjee, A.; Dhillon, I. S.; Ghosh, J.; Merugu, S.; and Modha, D. S. 2004a. A generalized maximum entropy approach to bregman co-clustering and matrix approximation. In *KDD*, 509–514.

Banerjee, A.; Merugu, S.; Dhillon, I. S.; and Ghosh, J. 2004b. Clustering with bregman divergences. In *SDM*.

Bui, T. N., and Jones, C. 1993. A heuristic for reducing fill-in in sparse matrix factorization. In *PPSC*, 445–452.

Chan, P. K.; Schlag, M. D. F.; and Zien, J. Y. 1993. Spectral k-way ratio-cut partitioning and clustering. In *DAC '93*, 749–754.

Collins, M.; Dasgupta, S.; and Reina, R. 2001. A generalizaionof principal component analysis to the exponential family. In *NIPS'01*.

Dhillon, I.; Guan, Y.; and Kulis, B. 2004. A unified view of kernel k-means, spectral clustering and graph cuts. Technical Report TR-04-25, University of Texas at Austin.

Dhillon, I.; Guan, Y.; and Kulis, B. 2005. A fast kernel-based multilevel algorithm for graph clustering. In *KDD '05*.

Dhillon, I. S. 2001. Co-clustering documents and words using bipartite spectral graph partitioning. In *KDD*, 269–274.

Ding, C. H. Q.; He, X.; Zha, H.; Gu, M.; and Simon, H. D. 2001. A min-max cut algorithm for graph partitioning and data clustering. In *Proceedings of ICDM 2001*, 107–114.

Hendrickson, B., and Leland, R. A multilevel algorithm for partitioning graphs. In *Supercomputing '95*.

Karypis, G., and Kumar, V. 1998. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.* 20(1):359–392.

Kernighan, B., and Lin, S. 1970. An efficient heuristic procedure for partitioning graphs. *The Bell System Technical Journal* 49(2):291–307.

Lang, K. 1995. News weeder: Learning to filter netnews. In *ICML*.

Long, B.; Wu, X.; Zhang, Z. M.; and Yu, P. S. 2006. Unsupervised learning on k-partite graphs. In *KDD-2006*.

Ng, A.; Jordan, M.; and Weiss, Y. 2001. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*.

S.D.Pietra, V.D.Pietera, J. 2001. Duality and auxiliary functions for bregman distances. Technical Report CMU-CS-01-109, Carnegie Mellon University.

Shi, J., and Malik, J. 2000. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(8):888–905.

Strehl, A., and Ghosh, J. 2002. Cluster ensembles – a knowledge reuse framework for combining partitionings. In *AAAI 2002*, 93–98.

Wang, S., and Schuurmans, D. 2003. Learning latent variable models with bregman divergences. In *IEEE International Symposium on Information Theory*.

Yu, K.; Yu, S.; and Tresp, V. 2005. Soft clustering on graphs. In *NIPS'05*.

Zha, H.; Ding, C.; Gu, M.; He, X.; and Simon, H. 2001. Bi-partite graph partitioning and data clustering. In *ACM CIKM'01*.

Zha, H.; Ding, C.; Gu, M.; He, X.; and Simon, H. 2002. Spectral relaxation for k-means clustering. *Advances in Neural Information Processing Systems* 14.

Relational Clustering by Symmetric Convex Coding

Bo Long BLONG1@BINGHAMTON.EDU
Zhongfei (Mark) Zhang ZZHANG@BINGHAMTON.EDU
Computer Science Dept., SUNY Binghamton, Binghamton, NY 13902

Xiaoyun Wu XIAOYUNW@GOOGLE.COM
Google Inc, 1600 Amphitheatre, Mountain View, CA 94043

Philip S. Yu PSYU@US.IBM.COM
IBM Watson Research Center, 19 skyline Drive, Hawthorne, NY 10532

Abstract

Relational data appear frequently in many machine learning applications. Relational data consist of the pairwise relations (similarities or dissimilarities) between each pair of implicit objects, and are usually stored in relation matrices and typically no other knowledge is available. Although relational clustering can be formulated as graph partitioning in some applications, this formulation is not adequate for general relational data. In this paper, we propose a general model for relational clustering based on symmetric convex coding. The model is applicable to all types of relational data and unifies the existing graph partitioning formulation. Under this model, we derive two alternative bound optimization algorithms to solve the symmetric convex coding under two popular distance functions, Euclidean distance and generalized I-divergence. Experimental evaluation and theoretical analysis show the effectiveness and great potential of the proposed model and algorithms.

1. Introduction

Two types of data are used in unsupervised learning, feature and relational data. Feature data are in the form of feature vectors and relational data consist of the pairwise relations (similarities or dissimilarities) between each pair of objects, and are usually stored in relation matrices and typically no other knowledge is available. Although feature data are the most common type of data, relational data have become more and more popular in many machine learning applications, such as web mining, social network analysis, bioinformatics, VLSI design, and task scheduling. Furthermore, the relational data are more general in the sense all the feature data can be transformed into relational data under a certain distance function.

The most popular way to cluster similarity-based relational data is to formulate it as the graph partitioning problem, which has been studied for decades. Graph partitioning seeks to cut a given graph into disjoint subgraphs which correspond to disjoint clusters based on a certain edge cut objective. Recently, graph partitioning with an edge cut objective has been shown to be mathematically equivalent to an appropriate weighted kernel k-means objective function (Dhillon et al., 2004; Dhillon et al., 2005). The assumption behind the graph partitioning formulation is that since the nodes within a cluster are similar to each other, they form a dense subgraph. However, in general this is not true for relational data, i.e., the clusters in relational data are not necessarily *dense* clusters consisting of strongly-related objects.

Figure 1 shows the relational data of four clusters, which are of two different types. In Figure 1, $\mathcal{C}_1 = \{v_1, v_2, v_3, v_4\}$ and $\mathcal{C}_2 = \{v_5, v_6, v_7, v_8\}$ are two traditional dense clusters within which objects are strongly related to each other. However, $\mathcal{C}_3 = \{v_9, v_{10}, v_{11}, v_{12}\}$ and $\mathcal{C}_4 = \{v_{13}, v_{14}, v_{15}, v_{16}\}$ also form two *sparse* clusters, within which the objects are not related to each other, but they are still "similar" to each other in the sense that they are related to the same set of other nodes. In Web mining, this type of cluster could be a group of music "fans" Web pages which share the same taste on the music and are linked to the same set of music Web pages but are not linked to each other (Kumar et al., 1999). Due to the importance of identifying this type of clusters (communities), it has been listed as one of the five algorithmic challenges in Web search engines (Henzinger et al., 2003). Note that the cluster structure of the relation data in Figure 1 cannot be correctly identified by graph partitioning approaches, since

Appearing in *Proceedings of the 24^{th} International Conference on Machine Learning*, Corvallis, OR, 2007.

they look for only dense clusters of strongly related objects by cutting the given graph into subgraphs; similarly, the pure bi-partite graph models cannot correctly identify this type of cluster structures. Note that re-defining the relations between the objects does not solve the problem in this situation, since there exist both dense and sparse clusters.

If the relational data are dissimilarity-based, to apply graph partitioning approaches to them, we need extra efforts on appropriately transforming them into similarity-based data and ensuring that the transformation does not change the cluster structures in the data. Hence, it is desirable for an algorithm to be able to identify the cluster structures no matter which type of relational data is given. This is even more desirable in the situation where the background knowledge about the meaning of the relations is not available, i.e., we are given only a relation matrix and do not know if the relations are similarities or dissimilarities.

In this paper, we propose a general model for relational clustering based on symmetric convex coding of the relation matrix. The proposed model is applicable to the general relational data consisting of only pairwise relations typically without other knowledge; it is capable of learning both dense and sparse clusters at the same time; it unifies the existing graph partition models to provide a generalized theoretical foundation for relational clustering. Under this model, we derive iterative bound optimization algorithms to solve the symmetric convex coding for two important distance functions, Euclidean distance and generalized I-divergence. The algorithms are applicable to general relational data and at the same time they can be easily adapted to learn a specific type of cluster structure. For example, when applied to learning only dense clusters, they provide new efficient algorithms for graph partitioning. The convergence of the algorithms is theoretically guaranteed. Experimental evaluation and theoretical analysis show the effectiveness and great potential of the proposed model and algorithms.

2. Related Work

Graph partitioning (or clustering) is a popular formulation of relational clustering, which divides the nodes of a graph into clusters by finding the best edge cuts of the graph. Several edge cut objectives, such as the average cut (Chan et al., 1993), average association (Shi & Malik, 2000), normalized cut (Shi & Malik, 2000), and min-max cut (Ding et al., 2001), have been proposed. Various spectral algorithms have been developed for these objective functions (Chan et al., 1993; Shi & Malik, 2000; Ding et al., 2001). These algorithms use the eigenvectors of a graph affinity matrix, or a matrix derived from the affinity matrix, to partition the graph.

Multilevel methods have been used extensively for graph partitioning with the Kernighan-Lin objective, which attempt to minimize the cut in the graph while maintaining equal-sized clusters (Bui & Jones, 1993; Hendrickson & Leland, 1995; Karypis & Kumar, 1998).

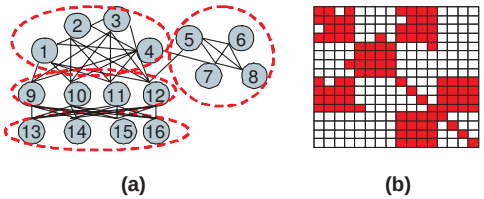

Figure 1. The graph (a) and relation matrix (b) of the relational data with different types of clusters. In (b), the dark color denotes 1 and the light color denotes 0.

Recently, graph partitioning with an edge cut objective has been shown to be mathematically equivalent to an appropriate weighted kernel k-means objective function (Dhillon et al., 2004; Dhillon et al., 2005). Based on this equivalence, the weighted kernel k-means algorithm has been proposed for graph partitioning (Dhillon et al., 2004; Dhillon et al., 2005). Yu et al. (2005) propose the graph-factorization clustering for the graph partitioning, which seeks to construct a bipartite graph to approximate a given graph. Nasraoui et al. (1999) propose the relational fuzzy maximal density estimator algorithm.

In this paper, our focus is on the homogeneous relational data, i.e., the objects in the data are of the same type. There are some efforts in the literature that can be considered as clustering heterogeneous relational data, i.e., different types of objects are related to each other. For example, co-clustering addresses clustering two types of related objects, such as documents and words, at the same time. Dhillon et al. (2003) propose a co-clustering algorithm to maximize the mutual information. A more generalized co-clustering framework is presented by Banerjee et al. (2004) wherein any Bregman divergence can be used in the objective function. Long et al. (2005), Li (2005) and Ding et al. (2006) all model the co-clustering as an optimization problem involving a triple matrix factorization.

3. Symmetric Convex Coding

In this section, we propose a general model for relational clustering. Let us first consider the relational data in Figure 1. An interesting observation is that although the different types of clusters look so different in the graph from Figure 1(a), they all demonstrate block patterns in the relation matrix of Figure 1(b) (without loss of generality, we arrange the objects from the same cluster together to make the block patterns explicit). Motivated by this observation, we propose the Symmetric Convex Coding (SCC) model to cluster relational data by learning the block pattern of a relation matrix. Since in most applications, the relations are of non-negative values and undirected, relational data can be represented as non-negative, symmetric matrices. Therefore, the definition of the SCC is given as follows.

Definition 3.1. Given a symmetric matrix $A \in \mathbb{R}_+$, a distance function $\mathfrak{D}$ and a positive number k, the symmetric

convex coding is given by the minimization,

$$\min_{\substack{C\in\mathbb{R}_+^{n\times k}, B\in\mathbb{R}_+^{k\times k}\\ C\mathbf{1}=\mathbf{1}}} \mathfrak{D}(A, CBC^T). \quad (1)$$

According to Definition 3.1, the elements of C are between 0 and 1 and the sum of the elements in each row of C equal to 1. Therefore, SCC seeks to use the convex combination of the *prototype matrix* B to approximate the original relation matrix. The factors from SCC have intuitive interpretations. The factor C is the soft membership matrix such that C_{ij} denotes the weight that the ith object associates with the jth cluster. The factor B is the prototype matrix such that B_{ii} denotes the connectivity within the ith cluster and B_{ij} denotes the connectivity between the ith cluster and the jth cluster.

SCC provides a general model to learn various cluster structures from relational data. Graph partitioning, which focuses on learning dense cluster structure, can be formulated as a special case of the SCC model. We propose the following theorem to show that the various graph partitioning objective functions are mathematically equivalent to a special case of the SCC model. Since most graph partitioning objective functions are based on the hard cluster membership, in the following theorem we modify the constraints on C as $C \in \mathbb{R}_+$ and $C^TC = I_k$ to make C to be the following cluster indicator matrix,

$$C_{ij} = \begin{cases} \frac{1}{|\pi_j|^{\frac{1}{2}}} & \text{if } v_i \in \pi_j \\ 0 & \text{otherwise} \end{cases}$$

where $|\pi_j|$ denotes the number of nodes in the jth cluster.

Theorem 3.2. *The hard version of SCC model under Euclidean distance function and $B = rI_k$ for $r > 0$, i.e.,*

$$\min_{\substack{C\in\mathbb{R}_+^{n\times k}, B\in\mathbb{R}_+^{k\times k}\\ C^TC=I_k}} ||A - C(rI_k)C^T||^2 \quad (2)$$

is equivalent to the maximization

$$\max tr(C^TAC), \quad (3)$$

where tr denotes the trace of a matrix.

Proof. Let L denote the objective function in Eq. 2.

$$\begin{aligned} L &= ||A - rCC^T||^2 && (4)\\ &= \text{tr}((A - rCC^T)^T(A - rCC^T)) && (5)\\ &= \text{tr}(A^TA) - 2r\text{tr}(CC^TA) + r^2\text{tr}(CC^TCC^T) && (6)\\ &= \text{tr}(A^TA) - 2r\text{tr}(C^TAC) + r^2k && (7)\end{aligned}$$

The above deduction uses the property of trace $\text{tr}(XY) = \text{tr}(YX)$. Since $\text{tr}(A^TA)$, r and k are constants, the minimization of L is equivalent to the maximization of $\text{tr}(C^TAC)$. The proof is completed. □

Theorem 3.2 states that with the prototype matrix B restricted to be of the form rI_k, SCC under Euclidean distance is reduced to the trace maximization in (3). Since various graph partitioning objectives, such as ratio association (Shi & Malik, 2000), normalized cut (Shi & Malik, 2000), ratio cut (Chan et al., 1993), and Kernighan-Lin objective (Kernighan & Lin, 1970), can be formulated as the trace maximization (Dhillon et al., 2004; Dhillon et al., 2005), Theorem 3.2 establishes the connection between the SCC model and the existing graph partitioning objective functions. Based on this connection, it is clear that the existing graph partitioning models make an implicit assumption for the cluster structure of the relational data, i.e., the clusters are not related to each other (the off-diagonal elements of B are zeroes) and the nodes within clusters are related to each other in the same way (the diagonal elements of B are r). This assumption is consistent with the intuition about the graph partitioning, which seeks to "cut" the graph into k separate subgraphs corresponding to the strongly-related clusters.

With Theorem 3.2 we may put other types of structural constraints on B to derive new graph partitioning models. For example, we fix B as a general diagonal matrix instead of rI_k, i.e, the model fixes the off-diagonal elements of B as zero and learns the diagonal elements of B. This is a more flexible graph partitioning model, since it allows the connectivity within different clusters to be different. More generally, we can use B to restrict the model to learn other types of the cluster structures. For example, by fixing diagonal elements of B as zeros, the model focuses on learning only spare clusters (corresponding to bi-partite or k-partite subgraphs), which are important for Web community learning (Kumar et al., 1999; Henzinger et al., 2003). In summary, the prototype matrix B not only provides the intuition for the cluster structure of the data, but also provides a simple way to adapt the model to learn specific types of cluster structures.

4. Algorithm Derivation

In this section, we derive efficient algorithms for the SCC model under two popular distance functions, Euclidean distance and generalized I-divergence.

4.1. Algorithm for SCC under Euclidean Distance

We derive an alternative optimization algorithm for SCC under Euclidean distance, i.e., the algorithm alternatively updates B and C until convergence.

First we fix B to update C. To deal with the constraint $C\mathbf{1} = \mathbf{1}$ efficiently, we transform it to a "soft" constraint by adding a penalty term, $\alpha||C\mathbf{1} - \mathbf{1}||^2$, to the objective function, where α is a positive constant. Therefore, we obtain the following optimization.

$$\min_{C\in\mathbb{R}_+^{n\times k}} ||A - CBC^T||^2 + \alpha||C\mathbf{1} - \mathbf{1}||^2. \quad (8)$$

The objective function in (8) is quartic with respect to C. We derive an efficient updating rule for C based on the bound optimization procedure (Salakhutdinov & Roweis, 2003; D.D.Lee & H.S.Seung, 1999). The basic idea is to construct an auxiliary function which is a convex upper bound for the original objective function based on the solution obtained from the previous iteration. Then, a new solution to the current iteration is obtained by minimizing this upper bound. The definition of the auxiliary function and a useful lemma (D.D.Lee & H.S.Seung, 1999) are quoted as follows.

Definition 4.1. $G(S, S^t)$ is an auxiliary function for $F(S)$ if $G(S, S^t) \geq F(S)$ and $G(S, S) = F(S)$.

Lemma 4.2. *If G is an auxiliary function, then F is non-increasing under the updating rule $S^{t+1} = \arg\min_S G(S, S^t)$.*

We propose an auxiliary function for C in the following theorem.

Lemma 4.3.

$$\begin{aligned} G(C,\tilde{C}) &= \sum_{ij}(A_{ij}+\frac{\alpha}{n}-2\sum_{gh}(A_{ij}\tilde{C}_{ig}B_{gh}\tilde{C}_{jh}(1+2\log C_{jh} \\ &\quad -2\log\tilde{C}_{jh})+\frac{\alpha}{nk}\tilde{C}_{jh}(1+\log C_{jh}-\log\tilde{C}_{jh}))+ \\ &\quad \sum_{gh}([\tilde{C}B\tilde{C}^T]_{ij}\tilde{C}_{ig}B_{gh}\tilde{C}_{jh}\frac{C_{jh}^4}{\tilde{C}_{jh}^4}+ \\ &\quad \frac{\alpha}{2nk}[\tilde{C}\mathbf{1}]_j\tilde{C}_{jh}(\frac{C_{jh}^4}{\tilde{C}_{jh}^4}+1))) \end{aligned}$$

is an auxiliary function for

$$F(C) = ||A - CBC^T||^2 + \alpha||C\mathbf{1} - \mathbf{1}||^2. \qquad (9)$$

Proof. For convenience, we let $\beta = \frac{\alpha}{nk}$.

$$\begin{aligned} F(C) &= \sum_{ij}((A_{ij}-\sum_{gh}C_{ig}B_{gh}C_{jh})^2+\beta\sum_{gh}(C_{jh}-1)^2) \\ &\leq \sum_{ij}(\sum_{gh}\frac{\tilde{C}_{ig}B_{gh}\tilde{C}_{jh}}{[\tilde{C}B\tilde{C}^T]_{ij}}(A_{ij}-\frac{[\tilde{C}B\tilde{C}^T]_{ij}}{\tilde{C}_{ig}B_{gh}\tilde{C}_{jh}}C_{ig}B_{gh}C_{jh})^2 \\ &\quad +\beta\sum_{gh}\frac{\tilde{C}_{jh}}{[\tilde{C}\mathbf{1}]_j}(\frac{[\tilde{C}\mathbf{1}]_j}{\tilde{C}_{jh}}C_{jh}-1)^2) \\ &= \sum_{ij}(A_{ij}-2\sum_{gh}A_{ij}C_{ig}B_{gh}C_{jh}+ \\ &\quad \sum_{gh}\frac{[\tilde{C}B\tilde{C}^T]_{ij}}{\tilde{C}_{ig}B_{gh}\tilde{C}_{jh}}C_{ig}^2B_{gh}^2C_{jh}^2+\beta\sum_{gh}\frac{[\tilde{C}\mathbf{1}]_j}{\tilde{C}_{jh}}C_{jh}^2 \\ &\quad -2\beta\sum_{gh}C_{jh}+k\beta) \\ &= \sum_{ij}(A_{ij}+k\beta-2\sum_{gh}(A_{ij}\tilde{C}_{ig}B_{gh}\tilde{C}_{jh}\frac{C_{ig}C_{jh}}{\tilde{C}_{ig}\tilde{C}_{jh}}+ \\ &\quad \beta\tilde{C}_{jh}\frac{C_{jh}}{\tilde{C}_{jh}})+\sum_{gh}([\tilde{C}B\tilde{C}^T]_{ij}\tilde{C}_{ig}B_{gh}\tilde{C}_{jh}\frac{C_{ig}^2C_{jh}^2}{\tilde{C}_{ig}^2\tilde{C}_{jh}^2} \\ &\quad +\beta[\tilde{C}\mathbf{1}]_j\tilde{C}_{jh}\frac{C_{jh}^2}{\tilde{C}_{jh}^2})) \\ &\leq \sum_{ij}(A_{ij}+k\beta-2\sum_{gh}(A_{ij}\tilde{C}_{ig}B_{gh}\tilde{C}_{jh}(1+\log C_{ig} \\ &\quad +\log C_{jh}-\log\tilde{C}_{ig}-\log\tilde{C}_{jh})+\beta\tilde{C}_{jh}(1+\log C_{jh}- \\ &\quad \log\tilde{C}_{jh}))+\sum_{gh}(\frac{1}{2}[\tilde{C}B\tilde{C}^T]_{ij}\tilde{C}_{ig}B_{gh}\tilde{C}_{jh}(\frac{C_{ig}^4}{\tilde{C}_{ig}^4}+\frac{C_{jh}^4}{\tilde{C}_{jh}^4}) \\ &\quad +\frac{1}{2}\beta[\tilde{C}\mathbf{1}]_j\tilde{C}_{jh}(\frac{C_{jh}^4}{\tilde{C}_{jh}^4}+1))) \\ &= \sum_{ij}(A_{ij}+k\beta-2\sum_{gh}(A_{ij}\tilde{C}_{ig}B_{gh}\tilde{C}_{jh}(1+2\log C_{jh} \\ &\quad -2\log\tilde{C}_{jh})+\beta\tilde{C}_{jh}(1+\log C_{jh}-\log\tilde{C}_{jh}))+ \\ &\quad \sum_{gh}([\tilde{C}B\tilde{C}^T]_{ij}\tilde{C}_{ig}B_{gh}\tilde{C}_{jh}\frac{C_{jh}^4}{\tilde{C}_{jh}^4}+ \\ &\quad \frac{1}{2}\beta[\tilde{C}\mathbf{1}]_j\tilde{C}_{jh}(\frac{C_{jh}^4}{\tilde{C}_{jh}^4}+1))) \end{aligned}$$

During the above deduction, we uses Jensen's inequality, convexity of the quadratic function and inequalities, $x^2 + y^2 \geq 2xy$ and $x \geq 1 + \log x$. □

The following theorem provides the updating rule for C.

Theorem 4.4. *The objective function $F(C)$ in Eq.*(9) *is nonincreasing under the updating rule,*

$$C = \tilde{C} \odot (\frac{A\tilde{C}B+\frac{\alpha}{2}}{\tilde{C}B\tilde{C}^T\tilde{C}B+\frac{\alpha}{2}\tilde{C}E})^{\frac{1}{4}} \qquad (10)$$

where $\tilde{C}$ denotes the solution from the previous iteration, E denotes a $k \times k$ matrix of 1's, $\odot$ denotes entry-wise product, and the division between two matrices is entry-wise division.

Proof. Based on Lemma 4.3, take the derivative of $G(C, \tilde{C})$ w.r.t. C_{jh} to obtain

$$\begin{aligned} \frac{\partial G(C,\tilde{C})}{\partial C_{jh}} &= \sum_i\sum_{gh}(-4A_{ij}\tilde{C}_{ig}B_{gh}\frac{\tilde{C}_{jh}}{C_{jh}}-2\frac{\alpha}{nk}\frac{\tilde{C}_{jh}}{C_{jh}} \\ &\quad +4[\tilde{C}B\tilde{C}^T]_{ij}\tilde{C}_{ig}B_{gh}\frac{C_{jh}^3}{\tilde{C}_{jh}^3} \\ &\quad +2\frac{\alpha}{nk}[\tilde{C}\mathbf{1}]_j\frac{C_{jh}^3}{\tilde{C}_{jh}^3}). \end{aligned}$$

Solve $\frac{\partial G(C,\tilde{C})}{\partial C_{jh}} = 0$ to obtain

$$C_{jh} = \tilde{C}_{jh}(\frac{\sum_i\sum_{gh}A_{ij}\tilde{C}_{ig}B_{gh}+\frac{\alpha}{2}}{\sum_i\sum_{gh}[\tilde{C}B\tilde{C}^T]_{ij}\tilde{C}_{ig}B_{gh}+\frac{\alpha}{2}[\tilde{C}\mathbf{1}]_j})^{\frac{1}{4}}$$

Formulate the above equation into the matrix form

$$C = \tilde{C} \odot (\frac{A\tilde{C}B+\frac{\alpha}{2}}{\tilde{C}B\tilde{C}^T\tilde{C}B+\frac{\alpha}{2}\tilde{C}E})^{\frac{1}{4}}$$

By Lemma 4.2, the proof is completed. □

Similarly, we present the following theorems to derive the updating rule for B.

Algorithm 1 SCC-ED algorithm

Input: A graph affinity matrix A and a positive integer k.
Output: A community membership matrix C and a community structure matrix B.
Method:

1: Initialize B and C.
2: **repeat**
3:
$$B = B \odot \frac{C^T AC}{C^T CBC^T C}.$$
4:
$$C = C \odot (\frac{ACB + \frac{\alpha}{2}}{CBC^T CB + \frac{\alpha}{2} CE})^{\frac{1}{4}}$$
5: **until** convergence

Lemma 4.5.

$$G(B, \tilde{B}) = \sum_{ij}(A_{ij} - 2\sum_{gh} A_{ij} C_{ig} B_{gh} C_{jh} + \sum_{gh} [C\tilde{B}C]_{ij} C_{ig} C_{jh} \frac{B_{gh}^2}{\tilde{B}_{gh}})$$

is an auxiliary function for

$$F(B) = ||A - CBC^T||^2. \quad (11)$$

Theorem 4.6. *The objective function $F(B)$ in Eq.*(11) *is nonincreasing under the updating rule*

$$B = \tilde{B} \odot \frac{C^T AC}{C^T C\tilde{B}C^T C}. \quad (12)$$

Following the way to prove Lemma 4.3 and Theorem 4.4, it is not difficult to prove the above theorems. We omit details here.

We call the algorithm as the SCC-ED algorithm, which is summarized in Algorithm 1. The implementation of SCC-ED is simple and it is easy to take advantage of the distributed computation for a very large data set. The complexity of the algorithm is $O(tn^2k)$ for t iterations and it can be further reduced for sparse data. The convergence of the SCC-ED algorithm is guaranteed by Theorems 4.4 and 4.6.

If the task is to learn the dense clusters from similarity-based relational data as the graph partitioning does, SCC-ED can achieve this task simply by fixing B as the identity matrix and updating only C by (10) until convergence. In other words, updating rule (10) itself provides a new and efficient graph partitioning algorithm, which is computationally more efficient than the popular spectral graph partitioning approaches which involve expensive eigenvector computation (typically $O(n^3)$) and the extra post-processing (Yu & Shi, 2003) on eigenvectors to obtain the clustering. Compared with the multi-level approaches such as METIS (Karypis & Kumar, 1998), this new algorithm does not restrict clusters to have an equal size.

Another advantage of the SCC-ED algorithm is that it is very easy for the algorithm to incorporate constraints on B to learn a specific type of cluster structures. For example, if the task is to learn the sparse clusters by constraining the diagonal elements of B to be zero, we can enforce this constraint simply by initializing the diagonal elements of B as zeros. Then, the algorithm automatically only updates the off-diagonal elements of B and the diagonal elements of B are 'locked' to zeros.

Yet another interesting observation about SCC-ED is that if we set $\alpha = 0$ to change the updating rule for C into the following,

$$C = \tilde{C} \odot (\frac{A\tilde{C}B}{\tilde{C}B\tilde{C}^T\tilde{C}B})^{\frac{1}{4}}, \quad (13)$$

the algorithm actually provides the symmetric conic coding. This has been touched in the literature as the symmetric case of non-negative factorizaion (Catral et al., 2004; Ding et al., 2005; Long et al., 2005). Therefore, SCC-ED under $\alpha = 0$ also provides a theoretically sound solution to the symmetric nonnegative matrix factorization.

4.2. Algorithm for SCC under Generalized I-divergence

Under the generalized I-divergence, the SCC objective function is given as follows,

$$D(A||CBC^T) = \sum_{ij}(A_{ij} \log \frac{A_{ij}}{[CBC^T]_{ij}} - A_{ij} + [CBC^T]_{ij}) \quad (14)$$

Similarly, we derive an alternative bound optimization algorithm for this objective function. First, we derive the updating rule for C and our task is the following optimization.

$$\min_{C \in \mathbb{R}_+^{n \times k}} D(A||CBC^T) + \alpha||C\mathbf{1} - \mathbf{1}||^2. \quad (15)$$

Then, the following theorems provide the updating rule for C.

Lemma 4.7.

$$G(C, \tilde{C}) = \sum_{ij}(A_{ij} \log A_{ij} - A_{ij} + \frac{\alpha}{n} + A_{ij} \sum_{gh} (\frac{\tilde{C}_{ig} B_{gh} \tilde{C}_{jh}}{[\tilde{C}B\tilde{C}^T]_{ij}} \log \frac{\tilde{C}_{ig}\tilde{C}_{jh}}{[\tilde{C}B\tilde{C}^T]_{ij}}) + \sum_{gh} ((\tilde{C}_{ig} B_{gh} \tilde{C}_{jh} + \frac{\alpha}{nk}[\tilde{C}\mathbf{1}]_j \tilde{C}_{jh}) \frac{C_{jh}^2}{\tilde{C}_{jh}^2}) - 2\sum_{gh} ((A_{ij} \frac{\tilde{C}_{ig} B_{gh} \tilde{C}_{jh}}{[\tilde{C}B\tilde{C}^T]_{ij}} + \frac{\alpha}{nk}\tilde{C}_{jh}) \log C_{jh})$$

$$-2\sum_{gh}\frac{\alpha}{nk}\tilde{C}_{jh}(1-\log\tilde{C}_{jh}))$$

is an auxiliary function for

$$F(C)=D(A||CBC^T)+\alpha||C\mathbf{1}-\mathbf{1}||^2. \quad (16)$$

Theorem 4.8. *The objective function $F(C)$ in Eq.*(16) *is nonincreasing under the updating rule,*

$$C_{jh} = \tilde{C}_{jh}\left(\frac{\sum_i \frac{A_{ij}[\tilde{C}B]_{ih}}{[\tilde{C}B\tilde{C}^T]_{ij}}+\alpha}{\sum_i[\tilde{C}B]_{ih}+\alpha[\tilde{C}\mathbf{1}]_j}\right)^{\frac{1}{2}} \quad (17)$$

where $\tilde{C}$ denotes the solution from the previous iteration.

The following theorems provide the updating rule for B.

Lemma 4.9.

$$\begin{aligned}G(B,\tilde{B}) = &\sum_{ij}(A_{ij}\log A_{ij}-A_{ij}+\sum_{gh}C_{ig}B_{gh}C_{jh}\\ &-A_{ij}\sum_{gh}\left(\frac{C_{ig}\tilde{B}_{gh}C_{jh}}{[C\tilde{B}C^T]_{ij}}(\log C_{ig}B_{gh}C_{jh}\right.\\ &\left.-\log\frac{C_{ig}\tilde{B}_{gh}C_{jh}}{[C\tilde{B}C^T]_{ij}}\right)))\end{aligned}$$

is an auxiliary function for

$$F(B)=D(A||CBC^T). \quad (18)$$

Theorem 4.10. *The objective function $F(B)$ in Eq.*(18) *is nonincreasing under the updating rule,*

$$B_{gh} = \tilde{B}_{gh}\frac{\sum_{ij}\frac{A_{ij}C_{ig}C_{jh}}{[C\tilde{B}C^T]_{ij}}}{\sum_{ij}C_{ig}C_{jh}} \quad (19)$$

where $\tilde{B}$ denotes the solution from the previous iteration.

Due to the space limit, we omit the proofs for the above theorems. We call the algorithm based on updating rule (17) and (19) as SCC-GI, which provides another new relational clustering algorithm. Similarly, when applied to the similarity-based relational data of dense clusters, SCC-GI provides another new and efficient graph partitioning algorithm.

5. Experimental Results

This section provides empirical evidence to show the effectiveness of the SCC model and algorithms in comparison with two representative graph partitioning algorithms, a spectral approach, Normalized Cut (NC) (Shi & Malik, 2000), and a multilevel algorithm, METIS (Karypis & Kumar, 1998).

Table 1. Summary of the synthetic relational data

Graph	Parameter	n	k
syn1	$\begin{bmatrix}0.5 & 0 & 0\\ 0 & 0.5 & 0\\ 0 & 0 & 0.5\end{bmatrix}$	900	3
syn2	$1-syn1$	900	3
syn3	$\begin{bmatrix}0 & 0.1 & 0.1\\ 0.1 & 0 & 0.2\\ 0.1 & 0.2 & 0\end{bmatrix}$	900	3
syn4	$[0,1]^{10\times 10}$	5000	10

5.1. Data Sets and Parameter Setting

The data sets used in the experiments include synthetic data sets with various cluster structures and real data sets based on various text data from the 20-newsgroups (Lang, 1995), WebACE and TREC (Karypis, 2002).

First, we use synthetic binary relational data to simulate relational data with different types of clusters such as dense clusters, sparse clusters and mixed clusters. All the synthetic relational data are generated based on Bernoulli distribution. The distribution parameters to generate the graphs are listed in the second column of Table 1 as matrices (true prototype matrices for the data). In a parameter matrix P, P_{ij} denotes the probability that the nodes in the ith cluster are connected to the nodes in the jth cluster. For example, in data syn3, the nodes in cluster 2 are connected to the nodes in cluster 3 with probability 0.2 and the nodes within a cluster are connected to each other with probability 0. Syn2 is generated by using 1 minus syn1. Hence, syn1 and syn2 can be viewed as a pair of similarity/dissimilarity data. Data syn4 has ten clusters mixing with dense clusters and sparse clusters. Due to the space limit, its distribution parameters are omitted here. Totally syn4 has 5000 nodes and about 2.1 million edges.

The graphs based on the text data have been widely used to test graph partitioning algorithms (Ding et al., 2001; Dhillon, 2001; Zha et al., 2001). Note that there also exist feature-based algorithms to directly cluster documents based on word features. However, in this study our focus is clustering based on relations instead of features. Hence graph clustering algorithms are used as comparisons. We use various data sets from the 20-newsgroups (Lang, 1995), WebACE and TREC (Karypis, 2002), which cover data sets of different sizes, different balances and different levels of difficulties. We construct relational data for each text data set such that objects (documents) are related to each other with cosine similarities between the term-frequency vectors. A summary of all the data sets to construct relational data used in this paper is shown in Table 2, in which n denotes the number of objects in the relational data, k denotes the number of true clusters, and *balance* denotes the size ratio of the smallest clusters to the largest clusters.

For the number of clusters k, we simply use the number of the true clusters. Note that how to choose the optimal number of clusters is a nontrivial model selection problem and beyond the scope of this paper. For performance measure,

Table 2. Summary of relational data based on text data sets.

Name	n	k	Balance	Source
tr11	414	9	0.046	TREC
tr23	204	6	0.066	TREC
NG17-19	1600	3	0.5	20-newsgroups
NG1-20	14000	20	1.0	20-newsgroups
k1b	2340	6	0.043	WebACE
hitech	2301	6	0.192	TREC
classic3	3893	3	0.708	MEDLINE/ CISI/CRANFILD

we elect to use the Normalized Mutual Information (NMI) (Strehl & Ghosh, 2002) between the resulting cluster labels and the true cluster labels, which is a standard way to measure the cluster quality. The final performance score is the average of ten runs.

5.2. Results and Discussion

Table 3 shows the NMI scores of the four algorithms on synthetic and real relational data. Each NMI score is the average of ten test runs and the standard deviation is also reported. We observe that although there is no single winner on all the data, for most data SCC algorithms perform better than or close to NC and METIS. Especially, SCC-GI provides the best performance on eight of the eleven data sets.

For the synthetic data syn1, almost all the algorithms provide perfect NMI score, since the data are generated with very clear dense cluster structures, which can be seen from the parameter matrix in Table 1. For data syn2, the dissimilarity version of syn1, we use exactly the same set of true cluster labels as that of syn1 to measure the cluster quality; the SCC algorithms still provide almost perfect NMI score; however, the METIS totally fails on syn2, since in syn2 the clusters have the form of sparse clusters, and based on the edge cut objective, METIS looks for only dense clusters. An interesting observation is that the NC algorithm does not totally fail on syn2 and in fact it provides a satisfactory NMI score. This is due to that although the original objective of the NC algorithm focuses on dense clusters (its objective function can be formulated as the trace maximization in Eq. (3)), after relaxing C to an arbitrary orthonormal matrix, what NC actually does is to embed cluster structures into the eigen-space and to discover them by post-processing the eigenvectors. Besides the dense cluster structures, sparse cluster structures could also have a good embedding in the eigen-space under a certain condition.

In data syn3, the relations within clusters are sparser than the relations between clusters, i.e., it also has sparse clusters, but the structure is more subtle than syn2. We observe that NC does not provide a satisfactory performance and METIS totally fails; in the mean time, SCC algorithms identify the cluster structure in syn3 very well. Data syn4 is a large relational data set of ten clusters consisting of four dense clusters and six sparse clusters; we observe that the SCC algorithms perform significantly better than NC and METIS on it, since they can identify both dense clusters and sparse clusters at the same time.

For the real data based on the text data sets, our task is to find dense clusters, which is consistent with the objectives of graph partitioning approaches. Overall, the SCC algorithms perform better than NC and METIS on the real data sets. Especially, SCC-ED provides the best performance in most data sets. The possible reasons for this are discussed as follows. First, the SCC model makes use of any possible block pattern in the relation matrices; on the other hand, the edge-cut based approaches focus on diagonal block patterns. Hence, the SCC model is more robust to heavily overlapping cluster structures. For example, for the difficult NG17-19 dataset, SCC algorithms do not totally fail as NC and METIS do. Second, since the edge weights from different graphs may have very different probabilistic distributions, popular Euclidean distance function, which corresponds to normal distribution assumption, are not always appropriate. By Theorem 3.2, edge-cut based algorithms are based on Euclidean distance. On the other hand, SCC-ED is based on generalized I-divergence corresponding to Poisson distribution assumption, which is more appropriate for graphs based on text data. Note that how to choose distance functions for specific graphs is non-trivial and beyond the scope of this paper. Third, unlike METIS, the SCC algorithms do not restrict clusters to have an equal size and hence they are more robust to unbalanced clusters.

In the experiments, we observe that SCC algorithms performs stably and rarely provides unreasonable solution, though like other algorithms SCC algorithms provide local optima to the NP-hard clustering problem. In the experiments, we also observe that the order of the actual running time for the algorithms is consistent with theoretical analysis in Section 4.1, i.e., METIS$<$SCC$<$NC. For example, in a test run on NG1-20, METIS, SCC-ED, SCC-GI and NC take 8.96, 11.4, 12.1 and 35.8 seconds, respectively. METIS is the best, since it is quasi-linear.

We also run the SCC-ED algorithm on the actor/actress graph based on IMDB movie data set for a case study of social network analysis. We formulate a graph of 20000 nodes, in which each node represents an actors/actresses and the edges denote collaboration between them. The number of the cluster is set to be 200. Although there is no ground truth for the clusters, we observe that the results consist of a large number of interesting and meaningful clusters, such as clusters of actors with a similar style and tight clusters of the actors from a movie or a movie serial. For example, Table 4 shows Community 121 consisting of 21 actors/actresses, which contains the actors/actresses in movies series "The Lord of Rings".

6. Conclusions

In this paper, we propose a general model for relational clustering based on symmetric convex coding of the relation matrix. The proposed model is applicable to the gen-

Table 3. NMI comparisons of NC, METIS, SCC-ED and SCC-GI algorithms

Data	NC	METIS	SCC-ED	SCC-GI
syn1	0.9652 ± 0.031	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$
syn2	0.8062 ± 0.52	0.000 ± 0.00	0.9038 ± 0.045	$\mathbf{0.9753 \pm 0.011}$
syn3	0.636 ± 0.152	0.115 ± 0.001	0.915 ± 0.145	$\mathbf{1.000 \pm 0.000}$
syn4	0.611 ± 0.032	0.638 ± 0.001	0.711 ± 0.043	$\mathbf{0.788 \pm 0.041}$
tr11	0.629 ± 0.039	0.557 ± 0.001	0.6391 ± 0.033	$\mathbf{0.661 \pm 0.019}$
tr23	0.276 ± 0.023	0.138 ± 0.004	$\mathbf{0.335 \pm 0.043}$	0.312 ± 0.099
NG17-19	0.002 ± 0.002	0.091 ± 0.004	0.1752 ± 0.156	$\mathbf{0.225 \pm 0.045}$
NG1-20	0.510 ± 0.004	$\mathbf{0.526 \pm 0.001}$	0.5041 ± 0.156	0.519 ± 0.010
k1b	0.546 ± 0.021	0.243 ± 0.000	0.537 ± 0.023	$\mathbf{0.591 \pm 0.022}$
hitech	0.302 ± 0.005	$\mathbf{0.322 \pm 0.001}$	0.319 ± 0.012	0.319 ± 0.018
classic3	0.621 ± 0.029	0.358 ± 0.000	0.642 ± 0.043	$\mathbf{0.822 \pm 0.059}$

Table 4. The members of cluster 121 in the actor graph

Cluster 121
Viggo Mortensen, Sean Bean, Miranda Otto, Ian Holm, Brad Dourif, Cate Blanchett, Ian McKellen ,Liv Tyler , David Wenham , Christopher Lee, John Rhys-Davies , Elijah Wood , Bernard Hill, Sean Astin, Dominic Monaghan, Andy Serkis, Karl Urban , Orlando Bloom , Billy Boyd ,John Noble, Sala Baker

eral relational data with various types of clusters and unifies the existing graph partitioning models. We derive iterative bound optimization algorithms to solve the symmetric convex coding for two important distance functions, Euclidean distance and generalized I-divergence. The algorithms are applicable to general relational data and at the same time they can be easily adapted to learn specific types of cluster structures. The convergence of the algorithms is theoretically guaranteed. Experimental evaluation shows the effectiveness and the great potential of the proposed model and algorithms.

Acknowledgement

We thank the anonymous reviewers for insightful comments. This work is supported in part by NSF (IIS-0535162), AFRL (FA8750-05-2-0284), and AFOSR (FA9550-06-1-0327).

References

Banerjee, A., Dhillon, I. S., Ghosh, J., Merugu, S., & Modha, D. S. (2004). A generalized maximum entropy approach to bregman co-clustering and matrix approximation. *KDD* (pp. 509–514).

Bui, T. N., & Jones, C. (1993). A heuristic for reducing fill-in in sparse matrix factorization. *PPSC* (pp. 445–452).

Catral, M., Han, L., Neumann, M., & Plemmons, R. (2004). On reduced rank nonnegative matrix factorization for symmetric nonnegative matrices. *Linear Algebra and Its Application*.

Chan, P. K., Schlag, M. D. F., & Zien, J. Y. (1993). Spectral k-way ratio-cut partitioning and clustering. *DAC '93* (pp. 749–754).

D.D.Lee, & H.S.Seung (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, *401*, 788–791.

Dhillon, I., Guan, Y., & Kulis, B. (2004). *A unified view of kernel k-means, spectral clustering and graph cuts* (Technical Report TR-04-25). University of Texas at Austin.

Dhillon, I., Guan, Y., & Kulis, B. (2005). A fast kernel-based multilevel algorithm for graph clustering. *KDD '05*.

Dhillon, I. S. (2001). Co-clustering documents and words using bipartite spectral graph partitioning. *KDD* (pp. 269–274).

Dhillon, I. S., Mallela, S., & Modha, D. S. (2003). Information-theoretic co-clustering. *KDD'03* (pp. 89–98).

Ding, C., He, X., & Simon, H. (2005). On the equivalence of nonnegative matrix factorization and spectral clustering. *SDM'05*.

Ding, C., Li, T., Peng, W., & Park, H. (2006). Orthogonal nonnegative matrix tri-factorizations for clustering. *kdd'06*.

Ding, C. H. Q., He, X., Zha, H., Gu, M., & Simon, H. D. (2001). A min-max cut algorithm for graph partitioning and data clustering. *Proceedings of ICDM 2001* (pp. 107–114).

Hendrickson, B., & Leland, R. (1995). A multilevel algorithm for partitioning graphs. *Supercomputing '95* (p. 28).

Henzinger, M., Motwani, R., & Silverstein, C. (2003). Challenges in web search engines. *Proc. of the 18th International Joint Conference on Artificial Intelligence* (pp. 1573–1579).

Karypis, G. (2002). A clustering toolkit.

Karypis, G., & Kumar, V. (1998). A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.*, *20*, 359–392.

Kernighan, B., & Lin, S. (1970). An efficient heuristic procedure for partitioning graphs. *The Bell System Technical Journal*, *49*, 291–307.

Kumar, R., Raghavan, P., Rajagopalan, S., & Tomkins, A. (1999). Trawling the Web for emerging cyber-communities. *Computer Networks (Amsterdam, Netherlands: 1999)*, *31*, 1481–1493.

Lang, K. (1995). News weeder: Learning to filter netnews. *ICML*.

Li, T. (2005). A general model for clustering binary data. *KDD'05*.

Long, B., Zhang, Z. M., & Yu, P. S. (2005). Co-clustering by block value decomposition. *KDD'05*.

Nasraoui, O., Krishnapuram, R., & Joshi, A. (1999). Relational clustering based on a new robust estimator with application to web mining. *NAFIPS 99*.

Salakhutdinov, R., & Roweis, S. (2003). Adaptive overrelaxed bound optimization methods. *ICML'03*.

Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*, 888–905.

Strehl, A., & Ghosh, J. (2002). Cluster ensembles – a knowledge reuse framework for combining partitionings. *AAAI 2002* (pp. 93–98).

Yu, K., Yu, S., & Tresp, V. (2005). Soft clustering on graphs. *NIPS'05*.

Yu, S., & Shi, J. (2003). Multiclass spectral clustering. *ICCV'03*..

Zha, H., Ding, C., Gu, M., He, X., & Simon, H. (2001). Bi-partite graph partitioning and data clustering. *ACM CIKM'01*.

A Probabilistic Framework for Relational Clustering

Bo Long
Computer Science Dept.
SUNY Binghamton
Binghamton, NY 13902
blong1@binghamton.edu

Zhongfei (Mark) Zhang
Computer Science Dept.
SUNY Binghamton
Binghamton, NY 13902
zzhang@binghamton.edu

Philip S. Yu
IBM Watson Research Center
19 Skyline Drive
Hawthorne, NY 10532
psyu@us.ibm.com

ABSTRACT

Relational clustering has attracted more and more attention due to its phenomenal impact in various important applications which involve multi-type interrelated data objects, such as Web mining , search marketing, bioinformatics, citation analysis, and epidemiology. In this paper, we propose a probabilistic model for relational clustering, which also provides a principal framework to unify various important clustering tasks including traditional attributes-based clustering, semi-supervised clustering, co-clustering and graph clustering. The proposed model seeks to identify cluster structures for each type of data objects and interaction patterns between different types of objects. Under this model, we propose parametric hard and soft relational clustering algorithms under a large number of exponential family distributions. The algorithms are applicable to relational data of various structures and at the same time unifies a number of stat-of-the-art clustering algorithms: co-clustering algorithms, the k-partite graph clustering, and semi-supervised clustering based on hidden Markov random fields.

Categories and Subject Descriptions: E.4 [**Coding and Information Theory**]:Data compaction and compression; H.3.3[**Information search and Retrieval**]:Clustering; I.5.3[**Pattern Recognition**]:Clustering.

General Terms: Algorithms.

Keywords: Clustering, Relational data, Relational clustering, Semi-supervised clustering, EM-algorithm, Bregman divergences, Exponential families.

1. INTRODUCTION

Most clustering approaches in the literature focus on "flat" data in which each data object is represented as a fixed-length attribute vector [38]. However, many real-world data sets are much richer in structure, involving objects of multiple types that are related to each other, such as documents and words in a text corpus, Web pages, search queries and Web users in a Web search system, and shops, customers, suppliers, shareholders and advertisement media in a marketing system.

KDD'07, August 12–15, 2007, San Jose, California, USA.

In general, relational data contain three types of information, attributes for individual objects, homogeneous relations between objects of the same type, heterogeneous relations between objects of different types. For example, for a scientific publication relational data set of papers and authors, the personal information such as affiliation for authors are attributes; the citation relations among papers are homogeneous relations; the authorship relations between papers and authors are heterogeneous relations. Such data violate the classic IID assumption in machine learning and statistics and present huge challenges to traditional clustering approaches. An intuitive solution is that we transform relational data into flat data and then cluster each type of objects independently. However, this may not work well due to the following reasons.

First, the transformation causes the loss of relation and structure information [14]. Second, traditional clustering approaches are unable to tackle influence propagation in clustering relational data, i.e., the hidden patterns of different types of objects could affect each other both directly and indirectly (pass along relation chains). Third, in some data mining applications, users are not only interested in the hidden structure for each type of objects, but also interaction patterns involving multi-types of objects. For example, in document clustering, in addition to document clusters and word clusters, the relationship between document clusters and word clusters is also useful information. It is difficult to discover such interaction patterns by clustering each type of objects individually.

Moreover, a number of important clustering problems, which have been of intensive interest in the literature, can be viewed as special cases of relational clustering. For example, graph clustering (partitioning) [7, 42, 13, 6, 20, 28] can be viewed as clustering on singly-type relational data consisting of only homogeneous relations (represented as a graph affinity matrix); co-clustering [12, 2] which arises in important applications such as document clustering and micro-array data clustering, can be formulated as clustering on bi-type relational data consisting of only heterogeneous relations. Recently, semi-supervised clustering [46, 4] has attracted significant attention, which is a special type of clustering using both labeled and unlabeled data. In section 5, we show that semi-supervised clustering can be formulated as clustering on singly-type relational data consisting of attributes and homogeneous relations.

Therefore, relational data present not only huge challenges to traditional unsupervised clustering approaches, but also great need for theoretical unification of various clustering tasks. In this paper, we propose a probabilistic model for relational clustering, which also provides a principal framework to unify various important clustering tasks includ-

ing traditional attributes-based clustering, semi-supervised clustering, co-clustering and graph clustering. The proposed model seeks to identify cluster structures for each type of data objects and interaction patterns between different types of objects. It is applicable to relational data of various structures. Under this model, we propose parametric hard and soft relational clustering algorithms under a large number of exponential family distributions. The algorithms are applicable to various relational data from various applications and at the same time unify a number of stat-of-the-art clustering algorithms: co-clustering algorithms, the k-partite graph clustering, Bregman k-means, and semi-supervised clustering based on hidden Markov random fields.

2. RELATED WORK

Clustering on a special case of relational data, bi-type relational data consisting of only heterogeneous relations, such as the word-document data, is called co-clustering or biclustering. Several previous efforts related to co-clustering are model based [22, 23]. Spectral graph partitioning has also been applied to bi-type relational data [11, 25]. These algorithms formulate the data matrix as a bipartite graph and seek to find the optimal normalized cut for the graph. Due to the nature of a bipartite graph, these algorithms have the restriction that the clusters from different types of objects must have one-to-one associations. Information-theory based co-clustering has also attracted attention in the literature. [12] proposes a co-clustering algorithm to maximize the mutual information between the clustered random variables subject to the constraints on the number of row and column clusters. A more generalized co-clustering framework is presented by [2] wherein any Bregman divergence can be used in the objective function. Recently, co-clustering has been addressed based on matrix factorization. [35] proposes an EM-like algorithm based on multiplicative updating rules.

Graph clustering (partitioning) clusters homogeneous data objects based on pairwise similarities, which can be viewed as homogeneous relations. Graph partitioning has been studied for decades and a number of different approaches, such as spectral approaches [7, 42, 13] and multilevel approaches [6, 20, 28], have been proposed. Some efforts [17, 43, 21, 21, 1] based on stochastic block modeling also focus on homogeneous relations.

Compared with co-clustering and homogeneous-relation-based clustering, clustering on general relational data, which may consist of more than two types of data objects with various structures, has not been well studied in the literature. Several noticeable efforts are discussed as follows. [45, 19] extend the the probabilistic relational model to the clustering scenario by introducing latent variables into the model; these models focus on using attribute information for clustering. [18] formulates star-structured relational data as a star-structured m-partite graph and develops an algorithm based on semi-definite programming to partition the graph. [34] formulates multi-type relational data as K-partite graphs and proposes a family of algorithms to identify the hidden structures of a k-partite graph by constructing a relation summary network to approximate the original k-partite graph under a broad range of distortion measures. The above graph-based algorithms do not consider attribute information.

Some efforts on relational clustering are based on inductive logic programming [37, 24, 31]. Base on the idea of mutual reinforcement clustering, [51] proposes a framework for clustering heterogeneous Web objects and [47] presents an approach to improve the cluster quality of interrelated data objects through an iterative reinforcement clustering process. There are no sound objective function and theoretical proof on the effectiveness and correctness (convergence) of the mutual reinforcement clustering. Some efforts [26, 50, 49, 5] in the literature focus on how to measure the similarities or choosing cross-relational attributes.

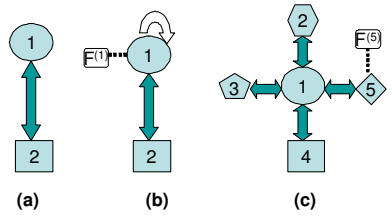

Figure 1: Examples of the structures of relational data.

To summarize, the research on relational data clustering has attracted substantial attention, especially in the special cases of relational data. However, there is still limited and preliminary work on general relational data clustering.

3. MODEL FORMULATION

With different compositions of three types of information, attributes, homogeneous relations and heterogeneous relations, relational data could have very different structures. Figure 1 shows three examples of the structures of relational data. Figure 1(a) refers to a simple bi-type of relational data with only heterogeneous relations such as word-document data. Figure 1(b) represents a bi-type data with all types of information, such as actor-movie data, in which actors (type 1) have attributes such as gender; actors are related to each other by collaboration in movies (homogeneous relations); actors are related to movies (type 2) by taking roles in movies (heterogeneous relations). Figure 1(c) represents the data consisting of companies, customers, suppliers, shareholders and advertisement media, in which customers (type 5) have attributes.

In this paper, we represent a relational data set as a set of matrices. Assume that a relational data set has m different types of data objects, $\mathcal{X}^{(1)} = \{x_i^{(1)}\}_{i=1}^{n_1}, \ldots, \mathcal{X}^{(m)} = \{x_i^{(m)}\}_{i=1}^{n_m}$, where n_j denotes the number of objects of the jth type and $x_p^{(j)}$ denotes the name of the pth object of the jth type. We represent the observations of the relational data as three sets of matrices, attribute matrices $\{\mathbf{F}^{(j)} \in \mathbb{R}^{d_j \times n_j}\}_{j=1}^{m}$, where d_j denotes the dimension of attributes for the jth type objects and $\mathbf{F}_{\cdot p}^{(j)}$ denotes the attribute vector for object $x_p^{(j)}$; homogeneous relation matrices $\{\mathbf{S}^{(j)} \in \mathbb{R}^{n_j \times n_j}\}_{j=1}^{m}$, where $\mathbf{S}_{pq}^{(j)}$ denotes the relation between $x_p^{(j)}$ and $x_q^{(j)}$; heterogeneous relation matrices $\{\mathbf{R}^{(ij)} \in \mathbb{R}^{n_i \times n_j}\}_{i,j=1}^{m}$, where $\mathbf{R}_{pq}^{(ij)}$ denotes the relation between $x_p^{(i)}$ and $x_q^{(j)}$. The above representation is a general formulation. In real applications, not every type of objects has attributes, homogeneous relations and heterogeneous relations. For example, the relational data set in Figure 1(a) is represented by only one heterogeneous matrix $\mathbf{R}^{(12)}$, and the one in Figure 1(b) is represented by three matrices, $\mathbf{F}^{(1)}$, $\mathbf{S}^{(1)}$ and $\mathbf{R}^{(12)}$. Moreover, for a specific clustering task, we may not use all available attributes and relations after feature or relation selection pre-processing.

Mixed membership models, which assume that each object has mixed membership denoting its association with

classes, have been widely used in the applications involving soft classification [16], such as matching words and pictures [39], race genetic structures [39, 48], and classifying scientific publications [15].

In this paper, we propose a relational mixed membership model to cluster relational data (we refer to the model as *mixed membership relational clustering* or MMRC throughout the rest of the paper).

Assume that each type of objects $\mathcal{X}^{(j)}$ has k_j latent classes. We represent the membership vectors for all the objects in $\mathcal{X}^{(j)}$ as a membership matrix $\Lambda^{(j)} \in [0,1]^{k_j \times n_j}$ such that the sum of elements of each column $\mathbf{\Lambda}_{\cdot p}^{(j)}$ is 1 and $\mathbf{\Lambda}_{\cdot p}^{(j)}$ denotes the membership vector for object $x_p^{(j)}$, i.e., $\mathbf{\Lambda}_{gp}^{(j)}$ denotes the probability that object $x_p^{(j)}$ associates with the gth latent class. We also write the parameters of distributions to generate attributes, homogeneous relations and heterogeneous relations in matrix forms. Let $\Theta^{(j)} \in \mathbb{R}^{d_j \times k_j}$ denote the distribution parameter matrix for generating attributes $\mathbf{F}^{(j)}$ such that $\Theta_{\cdot g}^{(j)}$ denotes the parameter vector associated with the gth latent class. Similarly, $\Gamma^{(j)} \in \mathbb{R}^{k_j \times k_j}$ denotes the parameter matrix for generating homogeneous relations $\mathbf{S}^{(j)}$; $\Upsilon^{(ij)} \in \mathbb{R}^{k_i \times k_j}$ denotes the parameter matrix for generating heterogeneous relations $\mathbf{R}^{(ij)}$. In summary, the parameters of MMRC model are

$$\Omega = \{\{\Lambda^{(j)}\}_{j=1}^m, \{\Theta^{(j)}\}_{j=1}^m, \{\Gamma^{(j)}\}_{j=1}^m, \{\Upsilon^{(ij)}\}_{i,j=1}^m\}.$$

In general, the meanings of the parameters, Θ, Λ, and Υ, depend on the specific distribution assumptions. However, in Section 4.1, we show that for a large number of exponential family distributions, these parameters can be formulated as expectations with intuitive interpretations.

Next, we introduce the latent variables into the model. For each object x_p^j, a latent cluster indicator vector is generated based on its membership parameter $\Lambda_{\cdot p}^{(j)}$, which is denoted as $\mathbf{C}_{\cdot p}^{(j)}$, i.e., $\mathbf{C}^{(j)} \in \{0,1\}^{k_j \times n_j}$ is a latent indicator matrix for all the jth type objects in $\mathcal{X}^{(j)}$.

Finally, we present the generative process of observations, $\{\mathbf{F}^{(j)}\}_{j=1}^m$, $\{\mathbf{S}^{(j)}\}_{j=1}^m$, and $\{\mathbf{R}^{(ij)}\}_{i,j=1}^m$ as follows:

1. For each object $x_p^{(j)}$
   - Sample $\mathbf{C}_{\cdot p}^{(j)} \sim Multinomial(\Lambda_{\cdot p}^{(j)}, 1)$.
2. For each object $x_p^{(j)}$
   - Sample $\mathbf{F}_{\cdot p}^{(j)} \sim Pr(\mathbf{F}_{\cdot p}^{(j)} | \Theta^{(j)} \mathbf{C}_{\cdot p}^{(j)})$.
3. For each pair of objects $x_p^{(j)}$ and $x_q^{(j)}$
   - Sample $\mathbf{S}_{pq}^{(j)} \sim Pr(\mathbf{S}_{pq}^{(j)} | (\mathbf{C}_{\cdot p}^{(j)})^T \Gamma^{(j)} \mathbf{C}_{\cdot q}^{(j)})$.
4. For each pair of objects $x_p^{(i)}$ and $x_q^{(j)}$
   - Sample $\mathbf{R}_{pq}^{(ij)} \sim Pr(\mathbf{R}_{pq}^{(ij)} | (\mathbf{C}_{\cdot p}^{(i)})^T \Upsilon^{(ij)} \mathbf{C}_{\cdot q}^{(j)})$.

In the above generative process, a latent indicator vector for each object is generated based on multinomial distribution with the membership vector as parameters. Observations are generated independently conditioning on latent indicator variables. The parameters of condition distributions are formulated as products of the parameter matrices and latent indicators, i.e., $Pr(\mathbf{F}_{\cdot p}^{(j)} | \mathbf{C}_{\cdot p}^{(j)}, \Theta^{(j)}) = Pr(F_{\cdot p}^{(j)} | \Theta^{(j)} \mathbf{C}_{\cdot p}^{(j)})$, $Pr(\mathbf{S}_{pq}^{(j)} | \mathbf{C}_{\cdot p}^{(j)}, \mathbf{C}_{\cdot q}^{(j)}, \Gamma^{(j)}) = Pr(\mathbf{S}_{pq}^{(j)} | (\mathbf{C}_{\cdot p}^{(j)})^T \Gamma^{(j)} \mathbf{C}_{\cdot q}^{(j)})$, and $Pr(\mathbf{R}_{pq}^{(ij)} | \mathbf{C}_{\cdot p}^{(i)}, \mathbf{C}_{\cdot q}^{(j)}, \Upsilon^{(ij)}) = Pr(\mathbf{R}_{pq}^{(ij)} | (\mathbf{C}_{\cdot p}^{(i)})^T \Upsilon^{(ij)} \mathbf{C}_{\cdot q}^{(j)})$. Under this formulation, an observation is sampled from the distributions of its associated latent classes. For example, if $\mathbf{C}_{\cdot p}^{(i)}$ indicates that $x_p^{(i)}$ is with the gth latent class and $\mathbf{C}_{\cdot q}^{(j)}$ indicates that $x_q^{(j)}$ is with the hth latent class, then $(\mathbf{C}_{\cdot p}^{(i)})^T \Upsilon^{(ij)} \mathbf{C}_{\cdot q}^{(j)} = \Upsilon_{gh}^{(ij)}$. Hence, we have $Pr(\mathbf{R}_{pq}^{(ij)} | \Upsilon_{gh}^{(ij)})$ implying that the relation between $x_p^{(i)}$ and $x_q^{(j)}$ is sampled by using the parameter $\Upsilon_{gh}^{(ij)}$.

With matrix representation, the joint probability distribution over the observations and the latent variables can be formulated as follows,

$$\begin{aligned} Pr(\Psi|\Omega) = \prod_{j=1}^m Pr(\mathbf{C}^{(j)}|\Lambda^{(j)}) \prod_{j=1}^m Pr(\mathbf{F}^{(j)}|\Theta^{(j)}\mathbf{C}^{(j)}) \\ \prod_{j=1}^m Pr(\mathbf{S}^{(j)}|(\mathbf{C}^{(j)})^T\Gamma^{(j)}\mathbf{C}^{(j)}) \\ \prod_{i=1}^m \prod_{j=1}^m Pr(\mathbf{R}^{(ij)}|(\mathbf{C}^{(i)})^T\Upsilon^{(ij)}\mathbf{C}^{(j)}) \end{aligned} \quad (1)$$

where $\Psi = \{\{\mathbf{C}^{(j)}\}_{j=1}^m, \{\mathbf{F}^{(j)}\}_{j=1}^m, \{\mathbf{S}^{(j)}\}_{j=1}^m, \{\mathbf{R}^{(ij)}\}_{i,j=1}^m\}$,
$Pr(\mathbf{C}^{(j)}|\Lambda^{(j)}) = \prod_{p=1}^{n_j} Multinomial(\Lambda_{\cdot p}^{(j)}, 1)$,
$Pr(\mathbf{F}^{(j)}|\Theta^{(j)}\mathbf{C}^{(j)}) = \prod_{p=1}^{n_j} Pr(\mathbf{F}_{\cdot p}^{(j)}|\Theta^{(j)}\mathbf{C}_{\cdot p}^{(j)})$,
$Pr(\mathbf{S}^{(j)}|(\mathbf{C}^{(j)})^T\Gamma^{(j)}\mathbf{C}^{(j)}) = \prod_{p,q=1}^{n_j} Pr(\mathbf{S}_{pq}^{(j)}|(\mathbf{C}_{\cdot p}^{(j)})^T\Gamma^{(j)}\mathbf{C}_{\cdot q}^{(j)})$,
and similarly for $\mathbf{R}^{(ij)}$.

4. ALGORITHM DERIVATION

In this section, based on the MMRC model we derive parametric soft and hard relational clustering algorithms under a large number of exponential family distributions.

4.1 MMRC with Exponential Families

To avoid clutter, instead of general relational data, we use relational data similar to the one in Figure 1(b), which is a representative relational data set containing all three types of information for relational data, attributes, homogeneous relations and heterogeneous relations. However, the derivation and algorithms are applicable to general relational data.

For the relational data set in Figure 1(b), we have two types of objects, one attribute matrix $\mathbf{F}$, one homogeneous relation matrix $\mathbf{S}$ and one heterogeneous relation matrix $\mathbf{R}$. Based on Eq.(1), we have the following likelihood function,

$$\begin{aligned} \mathcal{L}(\Omega|\Psi) = Pr(\mathbf{C}^{(1)}|\Lambda^{(1)})Pr(\mathbf{C}^{(2)}|\Lambda^{(2)})Pr(\mathbf{F}|\Theta\mathbf{C}^{(1)}) \\ Pr(\mathbf{S}|(\mathbf{C}^{(1)})^T\Gamma\mathbf{C}^{(1)})Pr(\mathbf{R}|(\mathbf{C}^{(1)})^T\Upsilon\mathbf{C}^{(2)}) \end{aligned} \quad (2)$$

Our goal is to maximize the likelihood function in Eq. (2) to estimate unknown parameters.

For the likelihood function in Eq.(2), the specific forms of condition distributions for attributes and relations depend on specific applications. Presumably, for a specific likelihood function, we need to derive a specific algorithm. However, a large number of useful distributions, such as normal distribution, Poisson distribution, and Bernoulli distributions, belong to exponential families and the distribution functions of exponential families can be formulated as a general form. This nice property facilitates us to derive a general EM algorithm for the MMRC model.

It is shown in the literature [3, 9] that there exists bijection between exponential families and Bregman divergences [40]. For example, the normal distribution, Bernoulli distribution, multinomial distribution and exponential distribution correspond to Euclidean distance, logistic loss, KL-divergence

and Itakura-Satio distance, respectively. Based on the bijection, an exponential family density $Pr(\mathbf{x})$ can always be formulated as the following expression with a Bregman divergence D_ϕ,

$$Pr(\mathbf{x}) = \exp(-D_\phi(\mathbf{x}, \mu))f_\phi(\mathbf{x}), \quad (3)$$

where $f_\phi(\mathbf{x})$ is a uniquely determined function for each exponential probability density, and μ is the expectation parameter. Therefore, for the MMRC model under exponential family distributions, we have the following,

$$Pr(\mathbf{F}|\Theta\mathbf{C}^{(1)}) = \exp(-D_{\phi_1}(\mathbf{F}, \Theta\mathbf{C}^{(1)}))f_{\phi_1}(\mathbf{F}) \quad (4)$$

$$Pr(\mathbf{S}|(\mathbf{C}^{(1)})^T\Gamma\mathbf{C}^{(1)}) = \exp(-D_{\phi_2}(\mathbf{S}, (\mathbf{C}^{(1)})^T\Gamma\mathbf{C}^{(1)}))f_{\phi_2}(\mathbf{S}) \quad (5)$$

$$Pr(\mathbf{R}|(\mathbf{C}^{(1)})^T\Upsilon\mathbf{C}^{(2)}) = \exp(-D_{\phi_3}(\mathbf{R}, (\mathbf{C}^{(1)})^T\Upsilon\mathbf{C}^{(2)}))f_{\phi_3}(\mathbf{R}) \quad (6)$$

In the above equations, a Bregman divergence of two matrices is defined as the sum of the Bregeman divergence of each pair of elements from the two matrices. Another advantage of the above formulation is that under this formulation, the parameters, Θ, Λ, and Υ, are expectations of intuitive interpretations. Θ consists of center vectors of attributes; Γ provides an intuitive summary of cluster structure within the same type objects, since $\Gamma^{(1)}_{gh}$ implies expectation relations between the gth cluster and the hth cluster of type 1 objects; similarly, Υ provides an intuitive summary for cluster structures between the different type objects. In the above formulation, we use different Bregman divergences, D_{ϕ_1}, D_{ϕ_2}, and D_{ϕ_3}, for the attributes, homogeneous relations and heterogeneous relations, since they could have different distributions in real applications. For example, suppose we have $\Theta^{(1)} = \begin{bmatrix} 1.1 & 2.3 \\ 1.5 & 2.5 \end{bmatrix}$ for normal distribution, $\Gamma^{(1)} = \begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.7 \end{bmatrix}$ for Bernoulli distribution , and $\Upsilon^{(12)} = \begin{bmatrix} 1 & 3 \\ 3 & 1 \end{bmatrix}$ for Poisson distribution, then the cluster structures of the data are very intuitive. First, the center attribute vectors for the two clusters of type 1 are $\begin{bmatrix} 1.1 \\ 1.5 \end{bmatrix}$ and $\begin{bmatrix} 2.3 \\ 2.5 \end{bmatrix}$; second, by $\Gamma^{(1)}$ we know that the type 1 nodes from different clusters are barely related and cluster 1 is denser that cluster 2; third, by $\Upsilon^{(12)}$ we know that cluster 1 of type 1 nodes are related to cluster 2 of type 2 nodes more strongly than to cluster 1 of type 2, and so on so forth.

Since the distributions of $\mathbf{C}^{(1)}$ and $\mathbf{C}^{(2)}$ are modeled as multinomial distributions, we have the following

$$Pr(\mathbf{C}^{(1)}|\Lambda^{(1)}) = \prod_{p=1}^{n_1}\prod_{g=1}^{k_1}(\Lambda^{(1)}_{gp})^{\mathbf{C}^{(1)}_{gp}} \quad (7)$$

$$Pr(\mathbf{C}^{(2)}|\Lambda^{(2)}) = \prod_{q=1}^{n_2}\prod_{h=1}^{k_2}(\Lambda^{(2)}_{hq})^{\mathbf{C}^{(2)}_{hq}} \quad (8)$$

Substituting Eqs. (4), (5), (6), (7), and (8) into Eq, (2) and taking some algebraic manipulations, we obtain the following log-likelihood function for MMRC under exponential families,

$$\log(\mathcal{L}(\Omega|\Psi)) = \sum_{p=1}^{n_1}\sum_{g=1}^{k_1}\mathbf{C}^{(1)}_{gp}\log\Lambda^{(1)}_{gp} + \sum_{q=1}^{n_2}\sum_{h=1}^{k_2}\mathbf{C}^{(2)}_{hq}\log\Lambda^{(2)}_{hq} - D_{\phi_1}(\mathbf{F}, \Theta\mathbf{C}^{(1)}) - D_{\phi_2}(\mathbf{S}, (\mathbf{C}^{(1)})^T\Gamma\mathbf{C}^{(1)}) - D_{\phi_3}(\mathbf{R}, (\mathbf{C}^{(1)})^T\Upsilon\mathbf{C}^{(2)}) + \tau \quad (9)$$

where $\tau = \log f_{\phi_1}(\mathbf{F}) + \log f_{\phi_2}(\mathbf{S}) + \log f_{\phi_3}(\mathbf{R})$, which is a constant in the log-likelihood function.

Expectation Maximization (EM) is a general approach to find the maximum-likelihood estimate of the parameters when the model has latent variables. EM does maximum likelihood estimation by iteratively maximizing the expectation of the complete (log-)likelihood, which is the following under the MMRC model,

$$\mathbf{Q}(\Omega, \tilde{\Omega}) = \mathbf{E}[\log(\mathcal{L}(\Omega|\Psi))|\mathbf{C}^{(1)}, \mathbf{C}^{(2)}, \tilde{\Omega}], \quad (10)$$

where $\tilde{\Omega}$ denotes the current estimation of the parameters and Ω is the new parameters that we optimize to increase $\mathbf{Q}$. Two steps, E-step and M-step, are alternatively performed to maximize the objective function in Eq. (10).

4.2 Monte Carlo E-step

In the E-step, based on Bayes's rule, the posterior probability of the latent variables,

$$Pr(\mathbf{C}^{(1)}, \mathbf{C}^{(2)}|\mathbf{F}, \mathbf{S}, \mathbf{R}, \tilde{\Omega}) = \frac{Pr(\mathbf{C}^{(1)}, \mathbf{C}^{(2)}, \mathbf{F}, \mathbf{S}, \mathbf{R}|\tilde{\Omega})}{\sum_{\mathbf{C}^{(1)}, \mathbf{C}^{(2)}} Pr(\mathbf{C}^{(1)}, \mathbf{C}^{(2)}, \mathbf{F}, \mathbf{S}, \mathbf{R}|\tilde{\Omega})}, \quad (11)$$

is updated using the current estimation of the parameters. However, conditioning on observations, the latent variables are not independent, i.e., there exist dependencies between the posterior probabilities of $\mathbf{C}^{(1)}$ and $\mathbf{C}^{(2)}$, and between those of $\mathbf{C}^{(1)}_{\cdot p}$ and $\mathbf{C}^{(1)}_{\cdot q}$. Hence, directly computing the posterior based on Eq. (11) is prohibitively expensive.

There exist several techniques for computing intractable posterior, such as Monte Carlo approaches, belief propagation, and variational methods. We follow a Monte Carlo approach, Gibbs sampler, which is a method of constructing a Markov chain whose stationary distribution is the distribution to be estimated.

It is easy to compute the posterior of a latent indicator vector while fixing all other latent indicator vectors, i.e.,

$$Pr(\mathbf{C}^{(1)}_{\cdot p}|\mathbf{C}^{(1)}_{\cdot -p}, \mathbf{C}^{(2)}, \mathbf{F}, \mathbf{S}, \mathbf{R}, \tilde{\Omega}) = \frac{Pr(\mathbf{C}^{(1)}, \mathbf{C}^{(2)}, \mathbf{F}, \mathbf{S}, \mathbf{R}|\tilde{\Omega})}{\sum_{\mathbf{C}^{(1)}_{\cdot p}} Pr(\mathbf{C}^{(1)}, \mathbf{C}^{(2)}, \mathbf{F}, \mathbf{S}, \mathbf{R}|\tilde{\Omega})}, \quad (12)$$

where $\mathbf{C}^{(1)}_{\cdot -p}$ denotes all the latent indicator vectors except for $\mathbf{C}^{(1)}_{\cdot p}$. Therefore, we present the following Markov chain to estimate the posterior in Eq. (11).

- Sample $\mathbf{C}^{(1)}_{\cdot 1}$ from distribution $Pr(\mathbf{C}^{(1)}_{\cdot 1}|\mathbf{C}^{(1)}_{\cdot -1}, \mathbf{C}^{(2)}, \mathbf{F}, \mathbf{S}, \mathbf{R}, \tilde{\Omega})$;
- $\cdots\cdots$
- Sample $\mathbf{C}^{(1)}_{\cdot n_1}$ from distribution $Pr(\mathbf{C}^{(1)}_{\cdot n_1}|\mathbf{C}^{(1)}_{\cdot -n_1}, \mathbf{C}^{(2)}, \mathbf{F}, \mathbf{S}, \mathbf{R}, \tilde{\Omega})$;
- Sample $\mathbf{C}^{(2)}_{\cdot 1}$ from distribution $Pr(\mathbf{C}^{(2)}_{\cdot 1}|\mathbf{C}^{(2)}_{\cdot -1}, \mathbf{C}^{(1)}, \mathbf{F}, \mathbf{S}, \mathbf{R}, \tilde{\Omega})$;
- $\cdots\cdots$
- Sample $\mathbf{C}^{(2)}_{\cdot n_2}$ from distribution $Pr(\mathbf{C}^{(2)}_{\cdot n_2}|\mathbf{C}^{(2)}_{\cdot -n_2}, \mathbf{C}^{(1)}, \mathbf{F}, \mathbf{S}, \mathbf{R}, \tilde{\Omega})$;

Note that at each sampling step in the above procedure, we use the latent indicator variables sampled from previous

steps. The above procedure iterates until the stop criterion is satisfied. It can be shown that the above procedure is a Markov chain converging to $Pr(\mathbf{C}^{(1)}, \mathbf{C}^{(2)}|\mathbf{F}, \mathbf{S}, \mathbf{R}, \tilde{\Omega})$. Assume that we keep l samples for estimation; then the posterior can be obtained simply by the empirical joint distribution of $\mathbf{C}^{(1)}$ and $\mathbf{C}^{(2)}$ in the l samples.

4.3 M-step

After the E-step, we have the posterior probability of latent variables to evaluate the expectation of the complete log-likelihood,

$$\mathbf{Q}(\Omega, \tilde{\Omega}) = \sum_{\mathbf{C}^{(1)}, \mathbf{C}^{(2)}} \log(\mathcal{L}(\Omega|\Psi)) Pr(\mathbf{C}^{(1)}, \mathbf{C}^{(2)}|\mathbf{F}, \mathbf{S}, \mathbf{R}, \tilde{\Omega}). \tag{13}$$

In the M-step, we optimize the unknown parameters by

$$\Omega^* = \arg\max_{\Omega} \mathbf{Q}(\Omega, \tilde{\Omega}). \tag{14}$$

First, we derive the update rules for membership parameters $\Lambda^{(1)}$ and $\Lambda^{(2)}$. To derive the expression for each $\mathbf{\Lambda}^{(1)}_{hp}$, we introduce the Lagrange multiplier α with the constraint $\sum_{g=1}^{k_1} \mathbf{\Lambda}^{(1)}_{gp} = 1$, and solve the following equation,

$$\frac{\partial}{\partial \mathbf{\Lambda}^{(1)}_{hp}} \{\mathbf{Q}(\Omega, \tilde{\Omega}) + \alpha(\sum_{g=1}^{k_1} \mathbf{\Lambda}^{(1)}_{gp} - 1)\} = 0. \tag{15}$$

Substituting Eqs. (9) and (13) into Eq. (15), after some algebraic manipulations, we have

$$Pr(\mathbf{C}^{(1)}_{hp} = 1|\mathbf{F}, \mathbf{S}, \mathbf{R}, \tilde{\Omega}) - \alpha \mathbf{\Lambda}^{(1)}_{hp} = 0. \tag{16}$$

Summing both sides over h, we obtain $\alpha = 1$ resulting in the following update rule,

$$\mathbf{\Lambda}^{(1)}_{hp} = Pr(\mathbf{C}^{(1)}_{hp} = 1|\mathbf{F}, \mathbf{S}, \mathbf{R}, \tilde{\Omega}), \tag{17}$$

i.e., $\mathbf{\Lambda}^{(1)}_{hp}$ is updated as the posterior probability that the pth object is associated with the hth cluster. Similarly, we have the following update rule for $\mathbf{\Lambda}^{(2)}_{hp}$

$$\mathbf{\Lambda}^{(2)}_{hp} = Pr(\mathbf{C}^{(2)}_{hp} = 1|\mathbf{F}, \mathbf{S}, \mathbf{R}, \tilde{\Omega}). \tag{18}$$

Second, we derive the update rule for Θ. Based on Eqs. (9) and (13), optimizing Θ is equivalent to the following optimization,

$$\arg\min_{\Theta} \sum_{\mathbf{C}^{(1)}, \mathbf{C}^{(2)}} D_{\phi_1}(\mathbf{F}, \Theta \mathbf{C}^{(1)}) Pr(\mathbf{C}^{(1)}, \mathbf{C}^{(2)}|\mathbf{F}, \mathbf{S}, \mathbf{R}, \tilde{\Omega}). \tag{19}$$

We reformulated the above expression as,

$$\arg\min_{\Theta} \sum_{\mathbf{C}^{(1)}} \sum_{g=1}^{k_1} \sum_{p:\mathbf{C}^{(1)}_{gp}=1} D_{\phi_1}(\mathbf{F}_{\cdot p}, \Theta_{\cdot g}) Pr(\mathbf{C}^{(1)}_{gp} = 1|\mathbf{F}, \mathbf{S}, \mathbf{R}, \tilde{\Omega}). \tag{20}$$

To solve the above optimization, we make use of an important property of Bregman divergence presented in the following theorem.

THEOREM 1. *Let X be a random variable taking values in $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n \subset S \subseteq \mathbb{R}^d$ following v. Given a Bregman divergence $D_\phi : S \times int(S) \mapsto [0, \infty)$, the problem*

$$\min_{s \in S} E_v[D_\phi(X, s)] \tag{21}$$

has a unique minimizer given by $s^* = E_v[X]$.

The proof of Theorem 1 is omitted (please refer [3, 40]). Theorem 1 states that the Bregman representative of a random variable is always the expectation of the variable. Based on Theorem 1 and the objective function in (20), we update $\Theta_{\cdot g}$ as follows,

$$\Theta_{\cdot g} = \frac{\sum_{p=1}^{n_1} \mathbf{F}_{\cdot p} Pr(\mathbf{C}^{(1)}_{gp} = 1|\mathbf{F}, \mathbf{S}, \mathbf{R}, \tilde{\Omega})}{\sum_{p=1}^{n_1} Pr(\mathbf{C}^{(1)}_{gp} = 1|\mathbf{F}, \mathbf{S}, \mathbf{R}, \tilde{\Omega})}. \tag{22}$$

Third, we derive the update rule for Γ. Based on Eqs. (9) and (13), we formulate optimizing Γ as the following optimization,

$$\arg\min_{\Gamma} \sum_{\mathbf{C}^{(1)}} \sum_{g=1}^{k_1} \sum_{h=1}^{k_1} \sum_{\substack{p:\mathbf{C}^{(1)}_{gp}=1, \\ q:\mathbf{C}^{(1)}_{hq}=1}} D_{\phi_2}(\mathbf{S}_{pq}, \Gamma_{gh}) \tilde{p}, \tag{23}$$

where $\tilde{p}$ denotes $Pr(\mathbf{C}^{(1)}_{gp} = 1, \mathbf{C}^{(1)}_{hq} = 1|\mathbf{F}, \mathbf{S}, \mathbf{R}, \tilde{\Omega})$ and $1 \leq p, q \leq n_1$. Based on Theorem 1, we update each Γ_{gh} as follows,

$$\Gamma_{gh} = \frac{\sum_{p,q=1}^{n_1} \mathbf{S}_{pq} Pr(\mathbf{C}^{(1)}_{gp} = 1, \mathbf{C}^{(1)}_{hq} = 1|\mathbf{F}, \mathbf{S}, \mathbf{R}, \tilde{\Omega})}{\sum_{p,q=1}^{n_1} Pr(\mathbf{C}^{(1)}_{gp} = 1, \mathbf{C}^{(1)}_{hq} = 1|\mathbf{F}, \mathbf{S}, \mathbf{R}, \tilde{\Omega})}. \tag{24}$$

Fourth, we derive the update rule for Υ. Based on Eqs. (9) and (13), we formulate optimizing Υ as the following optimization,

$$\arg\min_{\Upsilon} \sum_{\mathbf{C}^{(1)}, \mathbf{C}^{(2)}} \sum_{g=1}^{k_1} \sum_{h=1}^{k_2} \sum_{\substack{p:\mathbf{C}^{(1)}_{gp}=1, \\ q:\mathbf{C}^{(2)}_{hq}=1}} D_{\phi_3}(\mathbf{R}_{pq}, \Upsilon_{gh}) \tilde{p}, \tag{25}$$

where $\tilde{p}$ denotes $Pr(\mathbf{C}^{(1)}_{gp} = 1, \mathbf{C}^{(2)}_{hq} = 1|\mathbf{F}, \mathbf{S}, \mathbf{R}, \tilde{\Omega})$, $1 \leq p \leq n_1$ and $1 \leq q \leq n_2$. Based on Theorem 1, we update each Γ_{gh} as follows,

$$\Upsilon_{gh} = \frac{\sum_{p=1}^{n_1} \sum_{q=1}^{n_2} \mathbf{R}_{pq} Pr(\mathbf{C}^{(1)}_{gp} = 1, \mathbf{C}^{(2)}_{hq} = 1|\mathbf{F}, \mathbf{S}, \mathbf{R}, \tilde{\Omega})}{\sum_{p=1}^{n_1} \sum_{q=1}^{n_2} Pr(\mathbf{C}^{(1)}_{gp} = 1, \mathbf{C}^{(2)}_{hq} = 1|\mathbf{F}, \mathbf{S}, \mathbf{R}, \tilde{\Omega})}. \tag{26}$$

Combining the E-step and M-step, we have a general relational clustering algorithm, Exponential Family MMRC (EF-MMRC) algorithm, which is summarized in Algorithm 1. Since it is straightforward to apply our algorithm derivation to a relational data set of any structure, Algorithm 1 is proposed based on the input of a general relational data set. Despite that the input relational data could have various structures, EF-MMRC works simply as follows: in the E-step, EF-MMRC iteratively updates the posterior probabilities that an object is associated with the clusters (the Markov chain in Section 4.2); in the M-step, based on the current cluster association (posterior probabilities), the cluster representatives of attributes and relations are updated as the weighted mean of the observations no matter which exponential distributions are assumed.

Therefore, with the simplicity of the traditional centroid-based clustering algorithms, EF-MMRC is capable of making use of all attribute information and homogeneous and heterogenous relation information to learn hidden structures from various relational data. Since EF-MMRC simultaneously clusters multi-type interrelated objects, the cluster structures of different types of objects may interact with each other directly or indirectly during the clustering process to automatically deal with the influence propagation. Besides the local cluster structures for each type of objects,

Algorithm 1 Exponential Family MMRC Algorithm

Input: A relational data set $\{\{\mathbf{F}^{(j)}\}_{j=1}^m, \{\mathbf{S}^{(j)}\}_{j=1}^m, \{\mathbf{R}^{(ij)}\}_{i,j=1}^m\}$, a set of exponential family distributions (Bregman divergences) assumed for the data set.

Output: Membership Matrices $\{\Lambda^{(j)}\}_{j=1}^m$, attribute expectation matrices $\{\Theta^{(j)}\}_{j=1}^m$, homogeneous relation expectation matrices $\{\Gamma^{(j)}\}_{j=1}^m$, and heterogeneous relation expectation matrices $\{\Upsilon^{(ij)}\}_{i,j=1}^m$.

Method:

```
Initialize the parameters as Ω̃ = {{Λ̃^(j)}_{j=1}^m, {Θ̃^(j)}_{j=1}^m, {Γ̃^(j)}_{j=1}^m, {Υ̃^(ij)}_{i,j=1}^m}.
repeat
  {E-step}
  Compute the posterior Pr({C^(j)}_{j=1}^m | {F^(j)}_{j=1}^m, {S^(j)}_{j=1}^m, {R^(ij)}_{i,j=1}^m, Ω̃)
  using the Gibbs sampler.
  {M-step}
  for j = 1 to m do
    Compute Λ^(j) using update rule (17).
    Compute Θ^(j) using update rule (22).
    Compute Γ^(j) using update rule (24).
    for i = 1 to m do
      Compute Υ^(ij) using update rule (26).
    end for
  end for
  Ω̃ = Ω
until convergence
```

the output of EF-MMRC also provides the summary of the global hidden structure for the data, i.e., based on Γ and Υ, we know how the clusters of the same type and different types are related to each other. Furthermore, relational data from different applications may have different probabilistic distributions on the attributes and relations; it is easy for EF-MMRC to adapt to this situation by simply using different Bregman divergences corresponding to different exponential family distributions.

If we assume $O(m)$ types of heterogeneous relations among m types of objects, which is typical in real applications, and let $n = \Theta(n_i)$ and $k = \Theta(k_i)$, the computational complexity of EF-MMRC can be shown to be $O(tmn^2k)$ for t iterations. If we apply the k-means algorithm to each type of nodes individually by transforming the relations into attributes for each type of nodes, the total computational complexity is also $O(tmn^2k)$.

4.4 Hard MMRC Algorithm

Due to its simplicity, scalability, and broad applicability, k-means algorithm has become one of the most popular clustering algorithms. Hence, it is desirable to extend k-means to relational data. Some efforts [47, 2, 12, 33] in the literature work in this direction. However, these approaches apply to only some special and simple cases of relational data, such as bi-type heterogeneous relational data.

As traditional k-means can be formulated as a hard version of Gaussian mixture model EM algorithm [29], we propose the hard version of MMRC algorithm as a general relational k-means algorithm (from now on, we call Algorithm 1 as soft EF-MMRC), which applies to various relational data.

To derive the hard version MMRC algorithm, we omit soft membership parameters $\Lambda^{(j)}$ in the MMRC model ($\mathbf{C}^{(j)}$ in the model provides the hard membership for each object). Next, we change the computation of the posterior probabilities in the E-step to reassignment procedure, i.e., in the E-step, based on the estimation of the current parameters, we re-assign cluster labels, $\{\mathbf{C}^{(j)}\}_{j=1}^m$, to maximize the objective function in (9). In particular, for each object, while fixing the cluster assignments of all other objects, we assign it to each cluster to find the optimal cluster assignment maximizing the objective function in (9), which is equivalent to minimizing the Bregman distances between the observations and the corresponding expectation parameters. After all objects are assigned, the re-assignment process is repeated until no object changes its cluster assignment between two successive iterations.

In the M-step, we estimate the parameters based on the cluster assignments from the E-step. A simple way to derive the update rules is to follow the derivation in Section 4.3 but replace the posterior probabilities by its hard versions. For example, after the E-step, if the object $x_p^{(j)}$ is assigned to the gth cluster, i.e., $\mathbf{C}_{gp}^{(j)} = 1$, then the posterior $Pr(\mathbf{C}_{gp}^{(1)} = 1|\mathbf{F}, \mathbf{S}, \mathbf{R}, \tilde{\Omega}) = 1$ and $Pr(\mathbf{C}_{hp}^{(1)} = 1|\mathbf{F}, \mathbf{S}, \mathbf{R}, \tilde{\Omega}) = 0$ for $h \neq g$.

Using the hard versions of the posterior probabilities, we derive the following update rule for $\Theta^{(j)}$,

$$\Theta_{\cdot g}^{(j)} = \frac{\sum_{p:\mathbf{C}_{gp}^{(j)}=1} \mathbf{F}_{\cdot p}^{(j)}}{\sum_{p=1}^{n_j} \mathbf{C}_{gp}^{(j)}}. \tag{27}$$

In the above update rule, since $\sum_{p=1}^{n_1} \mathbf{C}_{gp}^{(j)}$ is the size of the gth cluster, $\Theta_{\cdot g}^{(j)}$ is actually updated as the mean of the attribute vectors of the objects assigned to the gth cluster.

Similarly, we have the following update rule for $\Gamma^{(j)}$

$$\Gamma_{gh}^{(j)} = \frac{\sum_{p:\mathbf{C}_{gp}^{(j)}=1, q:\mathbf{C}_{hq}^{(j)}=1} \mathbf{S}_{pq}^{(j)}}{\sum_{p=1}^{n_j} \mathbf{C}_{gp}^{(j)} \sum_{q=1}^{n_j} \mathbf{C}_{hq}^{(j)}}, \tag{28}$$

i.e., $\Gamma_{gh}^{(j)}$ is updated as the mean of the relations between the objects of the jth type from the gth cluster and from the hth cluster.

Each heterogeneous relation expectation parameter $\Upsilon_{gh}^{(ij)}$ is updated as the mean of the objects of the ith type from the gth cluster and of the jth type from the hth cluster,

$$\Upsilon_{gh}^{(ij)} = \frac{\sum_{p:\mathbf{C}_{gp}^{(i)}=1, q:\mathbf{C}_{hq}^{(j)}=1} \mathbf{R}_{pq}^{(ij)}}{\sum_{p=1}^{n_i} \mathbf{C}_{gp}^{(i)} \sum_{q=1}^{n_j} \mathbf{C}_{hq}^{(j)}}. \tag{29}$$

The hard version of EF-MMRC algorithm is summarized in Algorithm 2. It works simply as the classic k-means. However, it is applicable to various relational data under various Bregman distance functions corresponding to various assumptions of probability distributions. Based on the EM framework, its convergence is guaranteed. When applied to some special cases of relational data, it provides simple and new algorithms for some important data mining problems. For example, when applied to the data of one homogeneous relation matrix representing a graph affinity matrix, it provides a simple and new graph partitioning algorithm.

Based on Algorithms 1 and 2, there is another version of EF-MMRC, i.e., we may combine soft and hard EF-MMRC together to have mixed EF-MMRC. For example, we first run hard EF-MMRC several times as initialization, then run soft EF-MMRC.

Algorithm 2 Hard MMRC Algorithm

Input: A relational data set $\{\{\mathbf{F}^{(j)}\}_{j=1}^m, \{\mathbf{S}^{(j)}\}_{j=1}^m, \{\mathbf{R}^{(ij)}\}_{i,j=1}^m\}$, a set of exponential family distributions (Bregman divergences) assumed for the data set.
Output: Cluster indicator matrices $\{\mathbf{C}^{(j)}\}_{j=1}^m$, attribute expectation matrices $\{\Theta^{(j)}\}_{j=1}^m$, homogeneous relation expectation matrices $\{\Gamma^{(j)}\}_{j=1}^m$, and heterogeneous relation expectation matrices $\{\Upsilon^{(ij)}\}_{i,j=1}^m$.
Method:

```
Initialize the parameters as Ω̃ = {{Λ̃^(j)}_{j=1}^m, {Θ̃^(j)}_{j=1}^m, {Γ̃^(j)}_{j=1}^m, {Υ̃^(ij)}_{i,j=1}^m}.
repeat
  {E-step}
  Based on the current parameters, reassign cluster labels for each objects, i.e., update {C^(j)}_{j=1}^m, to maximize the objective function in Eq. (9).
  {M-step}
  for j = 1 to m do
    Compute Θ^(j) using update rule (27).
    Compute Γ^(j) using update rule (28).
    for i = 1 to m do
      Compute Υ^(ij) using update rule (29).
    end for
  end for
  Ω̃ = Ω
until convergence
```

5. A UNIFIED VIEW TO CLUSTERING

In this section we discuss the connections between existing clustering approaches and the MMRF model and EF-MMRF algorithms. By considering them as special cases or variations of the MMRF model, we show that MMRF provides a unified view to the existing clustering approaches from various important data mining applications.

5.1 Semi-supervised Clustering

Recently, semi-supervised clustering has become a topic of significant interest [4, 46], which seeks to cluster a set of data points with a set of pairwise constraints.

Semi-supervised clustering can be formulated as a special case of relational clustering, clustering on the single-type relational data set consisting of attributes $\mathbf{F}$ and homogeneous relations $\mathbf{S}$. For semi-supervised clustering, S_{pq} denotes the pairwise constraint on the pth object and the qth object.

[4] provides a general model for semi-supervised clustering based on Hidden Markov Random Fields (HMRFs). We show that it can be formulated as a special case of MMRC model. As in [4], we define the homogeneous relation matrix S as follows,

$$\mathbf{S}_{pq} = \begin{cases} f_M(x_p, x_q) & \text{if } (x_p, x_q) \in \mathcal{M} \\ f_C(x_p, x_q) & \text{if } (x_p, x_q) \in \mathcal{C} \\ 0 & \text{otherwise} \end{cases}$$

where $\mathcal{M}$ denotes a set of must-link constraints; $\mathcal{C}$ denotes a set of cannot-link constraints; $f_M(x_p, x_q)$ is a function that penalizes the violation of must-link constraint; $f_C(x_p, x_q)$ is a penalty function for cannot-links. If we assume Gibbs distribution [41] for $\mathbf{S}$,

$$Pr(S) = \frac{1}{z_1} \exp(-\sum_{p,q} \mathbf{S}_{pq}). \tag{30}$$

where z_1 is the normalization constant. Since [4] focuses on only hard clustering, we omit the soft member parameters in the MMRC model to consider hard clustering. Based on Eq.(30) and Eq.(4), the likelihood function of hard semi-supervised clustering under MMRC model is

$$L(\Theta|\mathbf{F}) = \frac{1}{z} \exp(-\sum_{p,q} \mathbf{S}_{pq}) \exp(-D_\phi(F, \Lambda\mathbf{C})) \tag{31}$$

Since $\mathbf{C}$ is an indicator matrix, Eq. (31) can be formulated as

$$L(\Theta|\mathbf{F}) = \frac{1}{z} \exp(-\sum_{p,q} \mathbf{S}_{pq}) \exp(-\sum_{g=1}^{k} \sum_{p:\mathbf{C}_{gp}=1} D_\phi(F_{\cdot p}, \Lambda_{\cdot g})) \tag{32}$$

The above likelihood function is equivalent to the objective function of semi-supervised clustering based on HMRFs [4]. Furthermore, when applied to optimizing the objective function in Eq.(32), hard MMRC provides a family of semi-supervised clustering algorithms similar to HMRF-KMeans in [4]; on the other hand, soft EF-MMRC provides new and soft version semi-supervised clustering algorithms.

5.2 Co-clustering

Co-clustering or bi-clustering arise in many important applications, such as document clustering, micro-array data clustering.A number of approaches [12, 8, 33, 2] have been proposed for co-clustering. These efforts can be generalized as solving the following matrix approximation problem [34],

$$\arg\min_{\mathbf{C},\Upsilon} \mathfrak{D}(\mathbf{R}, (\mathbf{C^{(1)}})^T \Upsilon \mathbf{C^{(2)}}) \tag{33}$$

where $R \in \mathbb{R}^{n_1 \times n_2}$ is the data matrix , $\mathbf{C}^{(1)} \in \{0,1\}^{k_1 \times n_1}$ and $\mathbf{C}^{(2)} \in \{0,1\}^{k_2 \times n_2}$ are indicator matrices, $\Upsilon \in \mathbb{R}^{k_1 \times k_2}$ is the relation representative matrix, and $\mathfrak{D}$ is a distance function. For example, [12] uses KL-divergences as the distance function; [8, 33] use Euclidean distances.

Co-clustering is equivalent to clustering on relational data of one heterogeneous relation matrix $\mathbf{R}$. Based on Eq.(9), by omitting the soft membership parameters, maximizing log-likelihood function of hard clustering on a heterogeneous relation matrix under the MMRC model is equivalent to the minimization in (33). The algorithms proposed in [12, 8, 33, 2] can be viewed as special cases of hard EF-MMRC. At the same time, soft EF-MMRC provides another family of new algorithms for co-clustering.

Our previous work [34] proposes the relation summary network model for clustering k-partite graphs, which can be shown to be equivalent on clustering on relational data of multiple heterogeneous relation matrices. The proposed algorithms in [34] can also be viewed as special cases of the hard EF-MMRC algorithm.

5.3 Graph Clustering

Graph clustering (partitioning) is an important problem in many domains, such as circuit partitioning, VLSI design, task scheduling. Existing graph partitioning approaches are mainly based on edge cut objectives, such as Kernighan-Lin objective [30], normalized cut [42], ratio cut [7], ratio association[42], and min-max cut [13].

Graph clustering is equivalent to clustering on single-type relational data of one homogeneous relation matrix $\mathbf{S}$. The log-likelihood function of the hard clustering under MMRC model is $-D_\phi(\mathbf{S}, (\mathbf{C})^T \Gamma \mathbf{C})$. We propose the following theorem to show that the edge cut objectives are mathematically equivalent to a special case of the MMRC model. Since most

graph partitioning objective functions use weighted indicator matrix such that $\mathbf{C}\mathbf{C}^T = \mathbf{I}_k$, where $\mathbf{I}_k$ is an identity matrix, we follow this formulation in the following theorem.

THEOREM 2. *With restricting Γ to be the form of $r\mathbf{I}_k$ for $r > 0$, maximizing the log-likelihood of hard MMRC clustering on $\mathbf{S}$ under normal distribution, i.e.,*

$$\max_{\mathbf{C}\in\{0,1\}^{k\times n}, \mathbf{C}\mathbf{C}^T=\mathbf{I}_k} -||\mathbf{S} - (\mathbf{C})^T(r\mathbf{I}_k)\mathbf{C}||^2, \qquad (34)$$

is equivalent to the trace maximization

$$\max tr(\mathbf{C}\mathbf{S}\mathbf{C}^T), \qquad (35)$$

where tr denotes the trace of a matrix.

PROOF. Let L denote the objective function in Eq. (34).

$$\begin{aligned} L &= -||\mathbf{S} - r\mathbf{C}^T\mathbf{C}||^2 \\ &= -\mathrm{tr}((\mathbf{S} - r\mathbf{C}^T\mathbf{C})^T(\mathbf{S} - r\mathbf{C}^T\mathbf{C})) \\ &= -\mathrm{tr}(\mathbf{S}^T\mathbf{S}) + 2r\mathrm{tr}(\mathbf{C}^T\mathbf{C}\mathbf{S}) - r^2\mathrm{tr}(\mathbf{C}^T\mathbf{C}\mathbf{C}^T\mathbf{C}) \\ &= -\mathrm{tr}(\mathbf{S}^T\mathbf{S}) + 2r\mathrm{tr}(\mathbf{C}\mathbf{S}\mathbf{C}^T) - r^2k \end{aligned}$$

The above deduction uses the property of trace $\mathrm{tr}(\mathbf{X}\mathbf{Y}) = \mathrm{tr}(\mathbf{Y}\mathbf{X})$. Since $\mathrm{tr}(\mathbf{S}^T\mathbf{S})$, r and k are constants, the maximization of L is equivalent to the maximization of $\mathrm{tr}(\mathbf{C}\mathbf{S}\mathbf{C}^T)$. The proof is completed. □

Since it is shown in the literature [10] that the edge cut objectives can be formulated as the trace maximization, Theorem 2 states that edge-cut based graph clustering is equivalent to MMRC model under normal distribution with the diagonal constraint on the parameter matrix Γ. This connection provides not only a new understanding for graph partitioning but also a family of new algorithms (soft and hard MMRC algorithms) for graph clustering.

Finally, we point out that MMRC model does not exclude traditional attribute-based clustering. When applied to an attribute data matrix under Euclidean distances, hard MMRC algorithm is actually reduced to the classic k-means; soft MMRC algorithm is very close to the traditional mixture model EM clustering except that it does not involve mixing proportions in the computation.

In summary, MMRC model provides a principal framework to unify various important clustering tasks including traditional attributes-based clustering, semi-supervised clustering, co-clustering and graph clustering; soft and hard EF-MMRC algorithms unify a number of stat-of-the-art clustering algorithms and at the same time provide new solutions to various clustering tasks.

6. EXPERIMENTS

This section provides empirical evidence to show the effectiveness of the MMRC model and algorithms. Since a number of stat-of-the-art clustering algorithms [12, 8, 33, 2, 3, 4] can be viewed as special cases of EF-MMRC model and algorithms, the experimental results in these efforts also illustrate the effectiveness of the MMRC model and algorithms. In this paper, we apply MMRC algorithms to tasks of graph clustering, bi-clustering, tri-clusering, and clustering on a general relational data set of all three types of information. In the experiments, we use mixed version MMRC, i.e., hard MMRC initialization followed by soft MMRC. Although MMRC can adopt various distribution assumptions, due to space limit, we use MMRC under normal or Poisson distribution assumption in the experiments. However, this does not imply that they are optimal distribution assumptions for the data. How to decide the optimal distribution assumption is beyond the scope of this paper.

Table 1: Summary of relational data for Graph Clustering.

Name	n	k	Balance	Source
tr11	414	9	0.046	TREC
tr23	204	6	0.066	TREC
NG1-20	14000	20	1.0	20-newsgroups
k1b	2340	6	0.043	WebACE

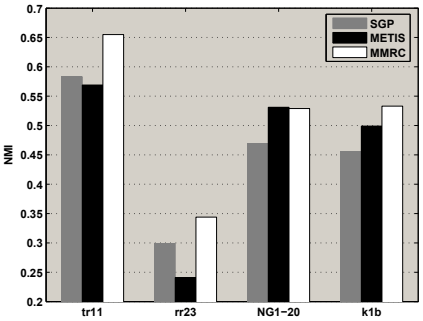

Figure 2: NMI comparison of SGP, METIS and MMRC algorithms.

For performance measure, we elect to use the Normalized Mutual Information (NMI) [44] between the resulting cluster labels and the true cluster labels, which is a standard way to measure the cluster quality. The final performance score is the average of ten runs.

6.1 Graph Clustering

In this section, we present experiments on the MMRC algorithm under normal distribution in comparison with two representative graph partitioning algorithms, the spectral graph partitioning (SGP) from [36] that is generalized to work with both normalized cut and ratio association, and the classic multilevel algorithm, METIS [28].

The graphs based on the text data have been widely used to test graph partitioning algorithms [13, 11, 25]. In this study, we use various data sets from the 20-newsgroups [32], WebACE and TREC [27], which cover data sets of different sizes, different balances and different levels of difficulties. The data are pre-processed by removing the stop words and each document is represented by a term-frequency vector using TF-IDF weights. Then we construct relational data for each text data set such that objects (documents) are related to each other with cosine similarities between the term-frequency vectors. A summary of all the data sets to construct relational data used in this paper is shown in Table 1, in which n denotes the number of objects in the relational data, k denotes the number of true clusters, and *balance* denotes the size ratio of the smallest clusters to the largest clusters.

For the number of clusters k, we simply use the number of the true clusters. Note that how to choose the optimal number of clusters is a nontrivial model selection problem and beyond the scope of this paper.

Figure 2 shows the NMI comparison of the three algorithms. We observe that although there is no single winner on all the graphs, overall the MMRC algorithm performs better than SGP and METIS. Especially on the difficult data set tr23, MMRC increases performance about 30%. Hence, MMRC under normal distribution provides a new graph partitioning algorithm which is viable and competi-

Data set	Taxonomy structure
TT-TM1	{rec.sport.baseball, rec.sport.hockey}, {talk.politics.guns, talk.politics.mideast, talk.politics.misc}
TT-TM2	{comp.graphics, comp.os.ms-windows.misc}, {rec.autos, rec.motorcycles}, {sci.crypt, sci.electronics}

Table 3: Taxonomy structures of two data sets for constructing tri-partite relational data

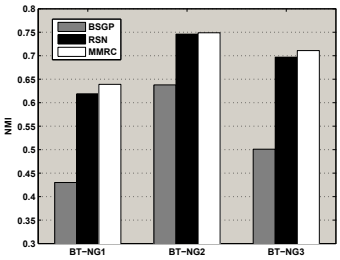

Figure 3: NMI comparison of BSGP, RSN and MMRC algorithms for bi-type data.

tive compared with the two existing state-of-the-art graph partitioning algorithms. Note that although the normal distribution is most popular, MMRC under other distribution assumptions may be more desirable in specific graph clustering applications depends on the statistical properties of the graphs.

6.2 Bi-clustering and Tri-clustering

In this section, we apply the MMRC algorithm under Poisson distribution to clustering bi-type relational data, word-document data, and tri-type relational data, word-document-category data. Two algorithms, Bi-partite Spectral Graph partitioning (BSGP) [11] and Relation Summary Network under Generalized I-divergence (RSN-GI) [34], are used as comparison in bi-clustering. For tri-clustering, Consistent Bipartite Graph Co-partitioning (CBGC) [18] and RSN-GI are used as comparison.

The bi-type relational data, word-document data, are constructed based on various subsets of the 20-Newsgroup data. We pre-process the data by selecting the top 2000 words by the mutual information. The document-word matrix is based on *tf.idf* weighting scheme and each document vector is normalized to a unit L_2 norm vector. Specific details of the data sets are listed in Table 2. For example, for the data set *BT-NG3* we randomly and evenly sample 200 documents from the corresponding newsgroups; then we formulate a bi-type relational data set of 1600 document and 2000 word.

The tri-type relational data are built based on the 20-newsgroups data for hierarchical taxonomy mining. In the field of text categorization, hierarchical taxonomy classification is widely used to obtain a better trade-off between effectiveness and efficiency than flat taxonomy classification. To take advantage of hierarchical classification, one must mine a hierarchical taxonomy from the data set. We see that words, documents, and categories formulate a sandwich structure tri-type relational data set, in which documents are central type nodes. The links between documents and categories are constructed such that if a document belongs to k categories, the weights of links between this document and these k category nodes are $1/k$ (please refer [18] for details). The true taxonomy structures for two data sets, *TP-TM1* and *TP-TM2*, are documented in Table 3.

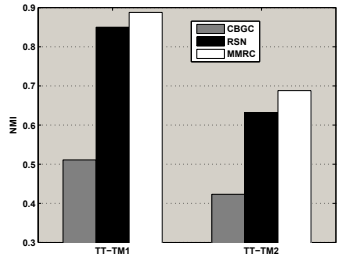

Figure 4: NMI comparison of CBGC, RSN and MMRC algorithms for tri-type data.

cluster 23 of actors
Viggo Mortensen, Sean Bean, Miranda Otto, Ian Holm, Christopher Lee, Cate Blanchett, Ian McKellen ,Liv Tyler , David Wenham , Brad Dourif , John Rhys-Davies , Elijah Wood , Bernard Hill , Sean Astin , Andy Serkis , Dominic Monaghan , Karl Urban , Orlando Bloom , Billy Boyd ,John Noble, Sala Baker
cluster 118 of movies
The Lord of the Rings: The Fellowship of the Ring (2001) The Lord of the Rings: The Two Towers (2002) The Lord of the Rings: The Return of the King (2003)

Table 4: Two clusters from actor-movie data

Figure 3 and Figure 4 show the NMI comparison of the three algorithms on bi-type and tri-type relational data, respectively. We observe that the MMRC algorithm performs significantly better than BSGP and CBGC. MMRC performs slightly better than RSN on some data sets. Since RSN is a special case of hard MMRC, this shows that mixed MMRC improves hard MMRC's performance on the data sets. Therefore, compared with the existing stated-of-the-art algorithms, the MMRC algorithm performs more effectively on these bi-clustering or tri-clustering tasks and on the other hand, it is flexible for different types of multi-clustering tasks which may be more complicated than tri-type clustering.

6.3 A Case Study on Actor-movie Data

We also run the MMRC algorithm on the actor-movie relational data based on IMDB movie data set for a case study. In the data, actors are related to each other by collaboration (homogeneous relations); actors are related to movies by taking roles in movies (heterogeneous relations); movies have attributes such as release time and rating (note that there is no links between movies). Hence the data have all the three types of information. We formulate a data set of 20000 actors and 4000 movies. We run experiments with $k = 200$. Although there is no ground truth for the data's cluster structure, we observe that most resulting clusters that are actors or movies of the similar style such as action, or tight groups from specific movie serials. For example, Table 4 shows cluster 23 of actors and cluster 118 of movies; the parameter $\Upsilon_{23,118}$ shows that these two clusters are strongly related to each other. In fact, the actor cluster contains the actors in the movie series "The Lord of the Rings". Note that if we only have one type of actor objects, we only get the actor clusters, but with two types of nodes, although there is no links between the movies, we also get the related movie clusters to explain how the actors are related.

7. CONCLUSIONS

In this paper, we propose a probabilistic model for relational clustering, which provides a principal framework to unify various important clustering tasks including tradi-

Dataset Name	Newsgroups Included	# Documents per Group	Total # Documents
BT-NG1	rec.sport.baseball, rec.sport.hockey	200	400
BT-NG2	comp.os.ms-windows.misc, comp.windows.x, rec.motorcycles, sci.crypt, sci.space	200	1000
BT-NG3	comp.os.ms-windows.misc, comp.windows.x, misc.forsale, rec.motorcycles,rec.motorcycles,sci.crypt, sci.space, talk.politics.mideast, talk.religion.misc	200	1600

Table 2: Subsets of Newsgroup Data for bi-type relational data

tional attributes-based clustering, semi-supervised clustering, co-clustering and graph clustering. Under this model, we propose parametric hard and soft relational clustering algorithms under a large number of exponential family distributions. The algorithms are applicable to relational data of various structures and at the same time unify a number of stat-of-the-art clustering algorithms. The theoretic analysis and experimental evaluation show the effectiveness and great potential of the proposed model and algorithms.

8. REFERENCES

[1] E. Airoldi, D. Blei, E. Xing, and S. Fienberg. Mixed membership stochastic block models for relational data with application to protein-protein interactions. In *ENAR-2006*.

[2] A. Banerjee, I. S. Dhillon, J. Ghosh, S. Merugu, and D. S. Modha. A generalized maximum entropy approach to bregman co-clustering and matrix approximation. In *KDD*, pages 509–514, 2004.

[3] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh. Clustering with bregman divergences. *J. Mach. Learn. Res.*, 6:1705–1749, 2005.

[4] S. Basu, M. Bilenko, and R. J. Mooney. A probabilistic framework for semi-supervised clustering. In *KDD04*, pages 59–68, 2004.

[5] I. Bhattachrya and L. Getor. Entity resolution in graph data. Technical Report CS-TR-4758, University of Maryland, 2005.

[6] T. N. Bui and C. Jones. A heuristic for reducing fill-in in sparse matrix factorization. In *PPSC*, pages 445–452, 1993.

[7] P. K. Chan, M. D. F. Schlag, and J. Y. Zien. Spectral k-way ratio-cut partitioning and clustering. In *DAC '93*.

[8] H. Cho, I. Dhillon, Y. Guan, and S. Sra. Minimum sum squared residue co-clustering of gene expression data. In *SDM*, 2004.

[9] M. Collins, S. Dasgupta, and R. Reina. A generalizaionof principal component analysis to the exponential family. In *NIPS'01*, 2001.

[10] I. Dhillon, Y. Guan, and B. Kulis. A unified view of kernel k-means, spectral clustering and graph cuts. Technical Report TR-04-25, University of Texas at Austin, 2004.

[11] I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *KDD'01*.

[12] I. S. Dhillon, S. Mallela, and D. S. Modha. Information-theoretic co-clustering. In *KDD'03*, pages 89–98.

[13] C. H. Q. Ding, X. He, H. Zha, M. Gu, and H. D. Simon. A min-max cut algorithm for graph partitioning and data clustering. In *Proceedings of ICDM 2001*, pages 107–114, 2001.

[14] S. Dzeroski and N. Lavrac, editors. *Relational Data Mining*. Springer, 2001.

[15] E. Erosheva, S. Fienberg, and J. Lafferty. Mixed membership models of scientific publications. In *NAS*.

[16] E. Erosheva and S. E. Fienberg. Bayesian mixed membership models for soft clustering and classification. *Classification-The Ubiquitous Challenge*, pages 11–26, 2005.

[17] S. E. Fienberg, M. M. Meyer, and S. Wasserman. Satistical analysis of multiple cociometric relations. *Journal of American Satistical Association*, 80:51–87, 1985.

[18] B. Gao, T.-Y. Liu, X. Zheng, Q.-S. Cheng, and W.-Y. Ma. Consistent bipartite graph co-partitioning for star-structured high-order heterogeneous data co-clustering. In *KDD '05*, pages 41–50, 2005.

[19] L. Getoor. An introduction to probabilistic graphical models for relational data. *Data Engineering Bulletin*, 29, 2006.

[20] B. Hendrickson and R. Leland. A multilevel algorithm for partitioning graphs. In *Supercomputing '95*, page 28, 1995.

[21] P. Hoff, A. Rafery, and M. Handcock. Latent space approaches to social network analysis. *Journal of American Satistical Association*, 97:1090–1098, 2002.

[22] T. Hofmann. Probabilistic latent semantic analysis. In *Proc. of Uncertainty in Artificial Intelligence, UAI'99*, Stockholm, 1999.

[23] T. Hofmann and J.Puzicha. Latent class models for collaborative filtering. In *IJCAI'99*, Stockholm, 1999.

[24] L. B. Holder and D. J. Cook. Graph-based relational learning: current and future directions. *SIGKDD Explor. Newsl.*, 5(1):90–93, 2003.

[25] M. X. H.Zha, C.Ding and H.Simon. Bi-partite graph partitioning and data clustering. In *ACM CIKM'01*, 2001.

[26] G. Jeh and J. Widom. Simrank: A measure of structural-context similarity. In *KDD-2002*, 2002.

[27] G. Karypis. A clustering toolkit, 2002.

[28] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.*, 20(1):359–392, 1998.

[29] M. Kearns, Y. Mansour, and A. Ng. An information-theoretic analysis of hard and soft assignment methods for clustering. In *UAI'97*, pages 282–293, 2004.

[30] B. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *The Bell System Technical Journal*, 49(2):291–307, 1970.

[31] M. Kirsten and S. Wrobel. Relational distance-based clustering. In *Proc. Fachgruppentreffen Maschinelles Lernen (FGML-98)*, pages 119 – 124, 1998.

[32] K. Lang. News weeder: Learning to filter netnews. In *ICML*, 1995.

[33] T. Li. A general model for clustering binary data. In *KDD'05*, 2005.

[34] B. Long, X. Wu, Z. M. Zhang, and P. S. Yu. Unsupervised learning on k-partite graphs. In *KDD-2006*, 2006.

[35] B. Long, Z. Zhang, and P. Yu. Co-clustering by block value decomposition. In *KDD'05*, 2005.

[36] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, 2001.

[37] L. D. Raedt and H. Blockeel. Using logical decision trees for clustering. In *Proceedings of the 7th International Workshop on Inductive Logic Programming*, 1997.

[38] R.O.Duda, P.E.Hart, and D.G.Stork. *Pattern Classification*. John Wiley & Sons, New York, 2000.

[39] N. Rosenberg, J. Pritchard, J. Weber, and H. Cann. Genetic structure of human population. *Science*, 298, 2002.

[40] J. S.D.Pietra, V.D.Pietera. Duality and auxiliary functions for bregman distances. Technical Report CMU-CS-01-109, Carnegie Mellon University, 2001.

[41] S.Geman and D.Geman. Stochastic relaxation, gibbs distribution, and the bayesian restoration of images. *Pattern Analysis and Machine Intelligence*, 6:721–742, 1984.

[42] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[43] T. Snijders. Markov chain monte carlo estimation of exponential random graph models. *Journal of Ssocial Structure*, 2002.

[44] A. Strehl and J. Ghosh. Cluster ensembles – a knowledge reuse framework for combining partitionings. In *AAAI 2002*, pages 93–98, 2002.

[45] B. Taskar, E. Segal, and D. Koller. Probabilistic classification and clustering in relational data. In *Proceeding of IJCAI-01*, 2001.

[46] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. Constrained k-means clustering with background knowledge. In *ICML-2001*, pages 577–584, 2001.

[47] J. Wang, H. Zeng, Z. Chen, H. Lu, L. Tao, and W.-Y. Ma. Recom: reinforcement clustering of multi-type interrelated data objects. In *SIGIR '03*, pages 274–281, 2003.

[48] E. Xing, A. Ng, M. Jorda, and S. Russel. Distance metric learning with applications to clustering with side information. In *NIPS'03*, volume 16, 2003.

[49] X. Yin, J. Han, and P. Yu. Cross-relational clustering with user's guidance. In *KDD-2005*, 2005.

[50] X. Yin, J. Han, and P. Yu. Linkclus: Efficient clustering via heterogeneous semantic links. In *VLDB-2006*, 2006.

[51] H.-J. Zeng, Z. Chen, and W.-Y. Ma. A unified framework for clustering heterogeneous web objects. In *WISE '02*, pages 161–172, 2002.